

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## HARDWARE PRO SBĚR TELEMETRICKÝCH DAT Z OSOBNÍHO VOZU

HARDWARE FOR CAR TELEMETRIC DATA COLLECTION

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Tomáš Musil

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Adam Ligocki

BRNO 2020

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Tomáš Musil

**ID:** 195394

**Ročník:** 3

**Akademický rok:** 2019/20

## NÁZEV TÉMATU:

### Hardware pro sběr telemetrických dat z osobního vozu

#### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a realizovat jednotku pro sběr telemetrických dat osobního vozu. Jednotka by měla mít podporu analogových vstupů a binárních vstupů a výstupů, RTC, podporu několika sběrnic pro připojení externích senzorů a rozhraní pro komunikace s externí vizualizační jednotkou.

1. Proveďte rešerši dostupných jednočipových mikrokontrolérů, které jsou adekvátní pro danou aplikaci a vyberte vhodnou platformu pro realizaci.
  2. Navrhněte DPS se zvoleným procesorem, která bude realizovat výše uvedené zadání sběru telemetrických dat.
  3. Navrhněte komunikační protokol pro výměnu dat s externí výpočetní jednotkou. Vytvořte zdroj uměle generovaných telemetrických dat pro testování bez HW.
  4. Realizujte DPS a otestujte funkčnost jednotlivých funkčních bloků plošného spoje.
  5. Realizujte software pro sběr dat z osobního vozu v reálném čase a odesílání do externí jednotky.
- Student při realizaci tohoto zadání bude spolupracovat s prací „Informační jednotka pro osobní vůz“

#### DOPORUČENÁ LITERATURA:

[1] Mazidi, Muhammad Ali, Shujen Chen, and Eshragh Ghaemi. STM32 Arm Programming for Embedded Systems (Volume 6). MicroDigitalEd. com, 2018.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

**Vedoucí práce:** Ing. Adam Ligocki

**doc. Ing. Václav Jirsík, CSc.**  
předseda rady studijního programu

#### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Cílem práce je navrhnout a realizovat jednotku pro sběr telemetrických dat z osobního vozu Škoda Felicia. Práce se nejprve zabývá rešerší dostupných jednočipovým mikrokontrolérů dostupných na trhu. Poté návrhem hardwaru s vybraným mikrokontrolérem a také vytvořením protokolu pro přenos dat z toho hardwaru do externí informační jednotky. Před samotnou realizací hardwaru, se práce zabývá vytvořením umělého zdroje dat pro testování navrženého protokolu. Posledním bodem je realizace hardwaru a následná implementace softwaru zajišťující sběr dat. Ten je umožněn částečně pomocí externích senzorů a také realizovanou komunikací s jednotkou osobního automobilu prostřednictvím komunikačního protokolu KW-1281.

## KLÍČOVÁ SLOVA

Návrh desky pro sběr dat, realizace desky pro sběr dat, návrh protokolu, umělý zdroj dat, software pro sběr dat, KWP-1281

## ABSTRACT

The aim of the work is to design and implement a unit for collecting telemetry data from a Škoda Felicia passenger car. The work first deals with a search of available single-chip microcontrollers available on the market. Then by designing the hardware with the selected microcontroller and also creating a protocol for transferring data from that hardware to an external information unit. Before the actual implementation of the hardware, the work deals with the creation of an artificial data source for testing the proposed protocol. The last point is the implementation of hardware and subsequent implementation of software providing data collection. This is made possible partly by external sensors and also by the implemented communication with the car unit via the communication protocol KW-1281.

## KEYWORDS

Design device for data collection, implementation of a board for data collection, design protocol, artificial data source, software for data collection, KWP-1281

MUSIL, Tomáš. *Hardware pro sběr telemetrických dat z osobního vozu*. Brno, 2020, 52 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: Ing. Adam Ligocki

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Hardware pro sběr telemetrických dat z osobního vozu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 8.6.2020

.....  
podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Adamu Ligockému za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno      8.6.2020

.....

podpis autora

# Obsah

<b>Úvod</b>	<b>10</b>
<b>1 Dostupné mikrokontroléry</b>	<b>11</b>
1.1 Mikrokontrolér . . . . .	11
1.2 Microchip-Atmel . . . . .	11
1.3 NXP . . . . .	12
1.4 STMicroelectronics . . . . .	12
1.4.1 STM32F407VGT6 . . . . .	12
1.4.2 STM32 Nucleo-144 . . . . .	13
1.4.3 STM32F407VET6 . . . . .	14
1.5 Volba mikrokontroléru . . . . .	15
<b>2 Hardware</b>	<b>16</b>
2.1 Koncept jednotky . . . . .	16
2.2 Mikrokontrolér STM32407VGT . . . . .	16
2.3 Zdroj energie pro jednotku . . . . .	17
2.4 Hodiny reálného času - RTC . . . . .	17
2.5 Digitální vstupy a výstupy . . . . .	17
2.6 Analogové vstupy . . . . .	18
2.7 Měření teploty . . . . .	19
2.8 Komunikace s externí jednotkou . . . . .	19
2.9 Průmyslová sběrnice RS-485 . . . . .	20
2.10 I2C . . . . .	20
2.11 CAN . . . . .	21
2.12 Vlastní návrh desky plošného spoje . . . . .	21
<b>3 Protokol a zdroj dat</b>	<b>22</b>
3.1 UART . . . . .	22
3.2 Protokol pro přenos dat . . . . .	22
3.2.1 Popis požadavku . . . . .	23
3.2.2 Popis odpovědi . . . . .	23
3.3 Umělý zdroj dat . . . . .	24
3.3.1 Popis aplikace . . . . .	25
<b>4 Realizace hardwaru</b>	<b>26</b>
4.1 Blokové schéma . . . . .	26
4.2 GPS . . . . .	26
4.3 Měření napětí autobaterie . . . . .	26

4.4	Detekce otevřených/zavřených dveří . . . . .	27
4.5	Měření doby vstřiku paliva . . . . .	28
4.5.1	Nastavení spojení . . . . .	28
4.5.2	Popis protokolu KW-1281 . . . . .	29
<b>5</b>	<b>Software</b>	<b>32</b>
5.1	STM32 Utility . . . . .	32
5.2	Inicializace zařízení . . . . .	32
5.3	Struktura hlavního programu . . . . .	32
5.4	Měření telemetrických hodnot . . . . .	33
5.5	Rychlost a ujetá vzdálenost . . . . .	33
5.6	Otevřené/zavřené dveře . . . . .	34
5.7	Napětí akumulátoru a sepnutí klíčku . . . . .	34
5.8	Měření teplot . . . . .	35
5.9	Výpočet aktuální a průměrné spotřeby paliva . . . . .	35
5.10	GPS . . . . .	36
5.11	Komunikace s externí jednotkou . . . . .	38
5.12	Testování . . . . .	40
<b>6</b>	<b>Závěr</b>	<b>41</b>
	<b>Literatura</b>	<b>42</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>45</b>
	<b>Seznam příloh</b>	<b>46</b>
<b>A</b>	<b>Schéma hardwarove desky pro sběr dat</b>	<b>47</b>
<b>B</b>	<b>Návrh PCB - Vrchní strana</b>	<b>48</b>
<b>C</b>	<b>Návrh PCB - Spodní strana</b>	<b>49</b>
<b>D</b>	<b>Realizovaná deska</b>	<b>50</b>
<b>E</b>	<b>Naskenovaný průběh odpovědi jednotka Bosch</b>	<b>51</b>
<b>F</b>	<b>Obsah přiloženého CD</b>	<b>52</b>



# Seznam obrázků

1.1	Microchip vývojová deska Xplained Pro s mikrokotrolérem ATSAM4E- XPRO . . . . .	11
1.2	STM32F407G-DISC1 . . . . .	13
1.3	STMicroelectronics Nucleo 144 . . . . .	13
1.4	STM32F407VET6 black board . . . . .	14
2.1	Použité piny mikrokontroléru . . . . .	16
2.2	Digitální vstupy a výstupy . . . . .	18
2.3	Analogový vstup . . . . .	19
2.4	Měření teploty . . . . .	19
2.5	Připojení externí jednotky . . . . .	20
3.1	UART diagram . . . . .	22
3.2	C# aplikace . . . . .	24
4.1	Blokove schéma znázorňující připojení komponent k jednotce . . . . .	26
4.2	Virace aplikace pro testování GPS knihovny a rozhraní . . . . .	27
4.3	Schéma dveřních kontaktů . . . . .	27
4.4	Diagnostický software VCDS . . . . .	31
5.1	ST-LINK UTILITY . . . . .	32
5.2	InOut jednotka . . . . .	39

# Seznam tabulek

1.1	Parametry mikrokontroléru ATSAM4E16EA . . . . .	11
1.2	Parametry mikrokontroléru MK20DN512ZVLL10 . . . . .	12
1.3	Parametry mikrokontroléru STM32F407VGT6 . . . . .	13
1.4	Parametry mikrokontroléru STM32F429ZIT6 . . . . .	14
1.5	Parametry mikrokontroléru STM32F407VET6 . . . . .	14
1.6	Přehled vlastností výše popsaných mikrokontrolerů . . . . .	15
2.1	Parametry DPS . . . . .	21
3.1	formát protokolu pro požadavek . . . . .	23
3.2	formát protokolu pro odpověď . . . . .	23
4.1	Nastavení spojení s řídicí jednotkou . . . . .	29
4.2	Příklad přijatých dat od řídicí jednotky . . . . .	30
4.3	Příklad ACK zprávy . . . . .	31
5.1	Data z ECU určeno výrobcem . . . . .	36
5.2	Struktura získaných a uložených dat . . . . .	37
5.3	GPS struktura získaných a uložených dat . . . . .	38
5.4	Souhrn implementovaných povelů . . . . .	39

# Úvod

Automobily v dnešní moderní době jsou čím dál tím více chytřejší, díky tomu se zlepšuje jejich funkcionalita. Dnes je naprosto běžné, že v každém moderním automobilu nalezneme nějaký druh více, či méně sofistikovaného palubního počítače. Takový palubní počítač je schopen měřit nespočet různých parametrů na základě, kterých může informovat řidiče o různých důležitých stavech vozu. Takovými informacemi mohou být například. průběžná spotřeba paliva, počet najetých kilometrů, servisní služby apod.. Stále se však po našich silnicích pohybuje mnoho automobilů, které touto vymožeností moderní doby nedisponují, ať už kvůli stáří vozu nebo kvůli nízké úrovni výbavy, ve které nejsou tyto počítače výrobcem dodávány.

Cílem této práce je navrhnout a realizovat jednotku do osobního automobilu Škoda Felicia, která by zajišťovala měření telemetrických dat při realizaci vlastního palubního počítače do tohoto automobilu. Telemetrickými daty se myslí například. vnitřní a venkovní teploty, spotřebované palivo, ujetá vzdálenost apod.. Samozřejmě se nabízí otázka, proč si nekoupit originální počítač. Zde ovšem nastává několik problémů, Felicie už přeci jen nejsou moderními vozy dnešní doby a díky tomu již není možné originální palubní počítač, jenž se do těchto vozů dodával, koupit. Dalším problémem je, že design těchto počítačů by již neodpovídal dnešním moderním představám takového počítače. Proto vznikla myšlenka realizovat vlastní počítač, jenž by odpovídal představám palubního počítače moderní doby se zachováním a také rozšířením funkcionalit starších verzí. Míra složitosti tohoto konceptu zapříčinila rozdělení práce na dvě části a to na část hardwaru pro sběr dat z osobního vozu, jenž získaná data distribuuje do druhé části, kterou je nadřazené zařízení pod názvem Informační jednotka pro osobní vůz. Informační jednotka je předmětem jiné práce. Tato práce se také kromě návrhu a realizace hardwaru zabývá vytvořením softwaru a návrhem protokolu, prostřednictvím kterého jsou naměřená data distribuována nadřazené jednotce.

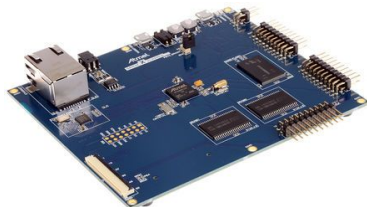
# 1 Dostupné mikrokontroléry

## 1.1 Mikrokontrolér

Jednočipový mikrokontrolér je obvod, jenž obsahuje kompletní mikropočítač. Jednočipové mikropočítače se vyznačují velkou spolehlivostí a kompaktností, proto jejich použití je většinou pro jednoúčelové aplikace jako je např: měření, řízení, regulace apod. A právě proto se tyto jednočipové počítače nejčastěji používají u embedded zařízení a systémů.[1] Pro naši aplikaci jsme využili mikrokontrolér s architekturou ARM. Při průzkumu trhu bylo zjištěno, že tento typ mikrokontrolérů vyrábí a dodává na trh několik firem z nichž nejznámější jsou Atmel,NXP, STMicroelectronics.

## 1.2 Microchip-Atmel

ATSAM4E je série založena na vysoce výkonném 32 bitovém ARM mikrokontroléru. Tento typ pracuje na maximálně frekvenci 120MHz. Obsahuje až 1024 kB Flash a 128kB SRAM paměti. Serie SAM4E nabízí bohatou sadu pokročilých periférií, které zahrnují např.Ethernet, CAN nebo podporu pro USB.[2] Konkrétní počet periférií a také velikost pamětí závisí na použití konkrétního typu mikrokontroléru z této řady. Příkladem použití je vývojová deska od Microchip Xplained Pro.



Obr. 1.1: Microchip vývojová deska Xplained Pro MCU ATSAM4E-XPRO [3]

Tab. 1.1: Xplained Pro parametry mikrokontroléru

ATSAM4E16EA pouzdro LQFP144	
Napájecí napětí	1,6V a 3,6V
Jádro	ARM Cortex M4
Maximální frekvence	120MHz
Paměť	1MB FLASH, 128kB SRAM
I/O porty	CAN(2x), UART(4x), I2C(2x),SPI(3x),Ethernet,...

## 1.3 NXP

Zástupcem od firmy NXP byl vybrán mikrokontrolér s označením MK20DN512ZVLL10, který patří do řady mikrokontrolerů K20. Jeho datová sběrnice má šířku 32bitů. Provozní napětí toho mikrokontroléru je 3,3V. Tento typ mikrokontroléru je jedním ze tří vybraných, kterými jsou MK10DN512ZVLL10 a MK60DN512ZVLL10. Tyto čipy se mezi sebou liší počtem vstupů, výstupů nebo dostupným typem a počtem rozhraní. Všechny ovšem obsahují periferie CAN a více než jeden UART díky tomu je možné použít pro naši aplikaci jakýkoliv z těchto tří typů.[4]

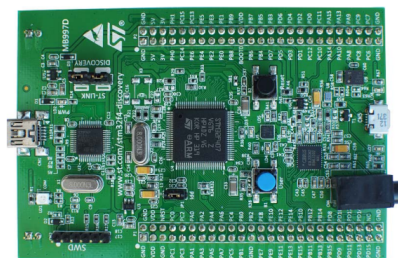
Tab. 1.2: Parametry mikrokontroléru MK20DN512ZVLL10

MK20DN512ZVLL10 pouzdro LQFP100	
Napájecí napětí	1,7V a 3,6V
Jádro	ARM Cortex M4
Maximální frekvence	100 MHz
Paměť	512 kB Flash, 128 kB RAM
Periferie	CAN(2x), SPI(3x), I2C(2x), UART(5x), USB

## 1.4 STMicroelectronics

### 1.4.1 STM32F407VGT6

STM32F407VGT6 je mikrokontrolér od STMicroelectronics. Jedná se o 32-bit ARM mikrokontrolér s jádrem Cortex-M4, který má v sobě zabudované dva typy paměti 1-MByte Flash a 192-Kbyte RAM. Tento chip může pracovat na maximální frekvenci 168MHz. Vývojové desky od firmy STM obsahují také debugovací nástroj ST-Link/V2 s funkcí virtuálního com portu. Výhodou tohoto nástroje, je že lze desku přímo spojit s vývojovým prostředím. Příkladem použití tohoto mikrokontroléru je Discovery Kit STM32F407G-DISC1 byl jedním z prvních od STM. Hlavní výhodou mikrokontrolerů a vývojových desek od STMicroelectronics je že firma dodává volný software včetně mnoha příkladů, což je skvělá pomůcka pro začátečníky. [5]



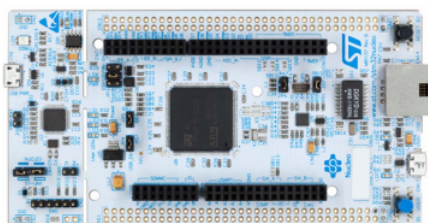
Obr. 1.2: STM32F407G Discovery kit s mikrokontrolérem STM32407VGT6 [6]

Tab. 1.3: Parametry mikrokontroléru STM32F407VGT6 [8]

STM32F407VGT6 pouzdro LQFP100	
Napájecí napětí	1,8V a 3,6V
Jádro	ARM Cortex M4
Maximální frekvence	168 MHz
Paměť	1024kB FLASH, 192kB RAM
Rozhraní	CAN(2x), I2C(3x), SDIO, SPI(3x), UART(2x), USART(4x), USB

### 1.4.2 STM32 Nucleo-144

STM32 Nucleo-144 je vývojová deska s mikrokontrolérem STM32F429ZIT6. Vývojových desek Nucleo je na trhu více variant, liší se zpravidla mikrokontrolérem a díky tomu i různým typem rozhraní. Desky Nucleo obsahují vysoce výkonné a úsporné mikrokontroléry, které jsou výborné pro tvorbu prototypů. Jejich výhodou je, že se na nich nachází ST Zio konektor, díky němuž je možné desky spojit s Arduino UnoV3. Jak už bylo zmíněno výše i zde je integrován programovací nástroj ST-LINK/V2 debugger. Samotný mikrokontrolér nabízí poměrně bohatou výbavu hardwaru. Pro naši aplikaci je nejdůležitější možnost práce se sběrnici CAN bus, a také 4 nezávislé periférie UART.[1]



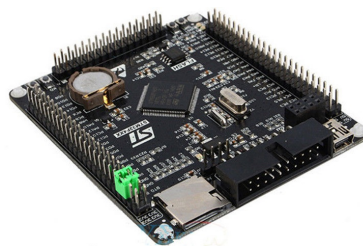
Obr. 1.3: STMicroelectronics Nucleo 144 [7]

Tab. 1.4: Parametry mikrokontroléru STM32F429ZIT6

STM32L4R5ZI pouzdro LQFP144	
Napájecí napětí	1,7V a 3,6V
Jádro	ARM Cortex M4
Maximální frekvence	180 MHz
Paměť	2 MB FLASH, 256 kB SRAM
Rozhraní	CAN(2x), I2C(3x), SPI(6x), UART(4x), USB OTG, atd..

### 1.4.3 STM32F407VET6

Mikrokontrolér STM32F407VET6 je jedním z rodiny mikrokontrolérů STM32F407xx. Možnostmi se velice podobá mikrokontroléru STM32F407VGT6. Tento mikrokontrolér je použit na vývojové desce black board, tato relativně malá vývojová deska umožňuje tvořit například aplikace s LCD displejem na který je vyveden konektor přímo na desce. Nachází se zde také pouzdro pro baterii jenž dodává napájení do RTC obvodu a slot pro MicroSD kartu.[9]



Obr. 1.4: STM32F407VET6 black board[10]

Tab. 1.5: STM32F407VET6

STM32F407VET6 pouzdro LQFP100	
Napájecí napětí	1,8V a 3,6V
Jádro	ARM Cortex M4
Maximální frekvence	168 MHz
Paměť	512 kB FLASH, 192 kB RAM
I/O porty	GPIO(82x), USART(4x), CAN(2x), SPI(3x), I2C(3x)

Tab. 1.6: Přehled vlastností výše popsanych mikrokontrolerů

Procesor	ATSAM4E16EA	MK20DN512Z	F407VGT6	F429ZIT6	F407VET6
Výrobce	Microchip	NXP	STM	STM	STM
Napětí	1,6 až 3,6V	1,7 až 3,6V	1,8 až 3,6V	1,7 až 3,6V	1,7 až 3,6V
Jádro	Cortex M4	Cortex M4	Cortex M4	Cortex M4	Cortex M4
Pouzdro	LQFP144	LQFP100	LQFP100	LQFP144	LQFP100
Kmitočet	120MHz	100MHz	168MHz	180MHz	168MHz
FLASH	1MB	512kB	1MB	2MB	512kB
RAM	SRAM 128kB	128kB	192kB	256kB	192kB
Periférie	CAN(2x) UART(4x) I2C(2x) SPI(3x) Ethernet	CAN(2x) SPI(3x) I2C(2x) UART(5x) USB	CAN(2x) I2C(3x) SDIO SPI(3x) UART(4x) USB	CAN(2x) I2C(3x) SPI(6x) UART(4x) SDIO USB OTG	CAN(2x) GPIO(82x) USART(4x) SPI(3x) I2C(3x)
Cena	258Kč	415Kč	329Kč	569Kč	268Kč

## 1.5 Volba mikrokontroléru

Při průzkumu trhu bylo zjištěno, že je možné zvolit podle našich požadavků, více typů mikrokontrolerů od různých výrobců. Díky této skutečnosti byl na základě předchozích zkušeností zvolen mikrokontrolér od firmy STMicroelectronics. Konkrétně mikrokontrolér s označením STM32F407VGT. Výhodou této volby je, že STMicroelectronics nabízí volně dostupné knihovny, díky kterým je práce s těmito mikrokontroléry snazší.





## 2.3 Zdroj energie pro jednotku

Jednotka je napájena ze stabilizátoru LM2596-ADJ, jehož vstup je připojen přes relé k 12V z baterie a toto napětí je přerušeno přes klíček od zapalování. Schéma pro zapojení stabilizátoru je částečně převzato z datasheetu od výrobce. Princip celého zapojení je navržen takto, jakmile dojde k zapnutí klíčku, mikrokontrolér sepne pomocí tranzistoru relé a tím dojde k přepnutí napájení ze spínaných 12V na stálých 12V, které nejsou závislé na klíčku. Aby bylo možné detekovat vypnutí zapalování je spínané napětí přivedeno na analogový vstup a v případě, že dojde k vypnutí zapalování mikrokontrolér uloží naměřená data a provede odpojení jednotky od napájení rozepnutím relé. Vzhledem k tomu, že mikrokontrolér ARM vyžaduje napájecí napětí 3,3V, bylo nutné také realizovat zdroj, který toto zajistí. Tento zdroj je realizován pomocí spínaného stabilizátoru LM2736. Protože jednotka bude pracovat ve velkém rozsahu teplot, byla zvolena reference LM4040 která zaručuje, že referenční napětí v daném rozsahu teplot bude téměř konstantní. Teplotní rozsah reference je od -40 do 85°C.

## 2.4 Hodiny reálného času - RTC

Obvod RTC slouží k udržení informací o reálném času a datu. Obvod disponuje svým vlastním krystalem, který musí být velmi přesný, aby nedocházelo k diferencí od reálného času. Obvod RTC se nachází uvnitř mikrokontroléru jako periferie, která běží nezávisle. Informace o reálném času se používají například při ukládání systémových záznamů na mikro SD kartu. Aby bylo možné udržet informaci musí obvod neustále běžet a k tomu potřebuje energii. Tuto energii mu dodává v případě zapnutého napájení stabilizátor a pokud dojde k vypnutí napájení je zde akumulátor energie v podobě kondenzátoru. Schéma viz příloha.

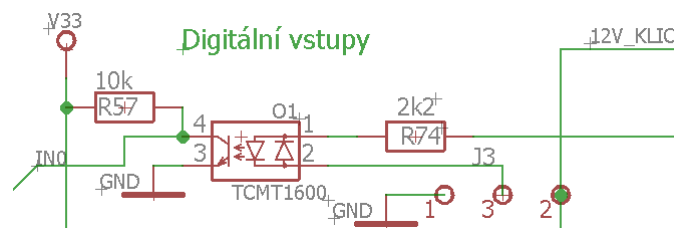
## 2.5 Digitální vstupy a výstupy

Digitální vstupy a výstupy mohou být využity ke komunikaci mezi obvody, ale také k měření veličin jako je např. rychlost vozu nebo počtu najetých kilometrů apod. U jednotlivých pinů mikrokontroléru musí být definováno, zda budou sloužit jako vstupy či výstupy. U digitálních pinů je také nutné definovat, zda mají reagovat na náběžnou, sestupnou nebo na úroveň signálu.

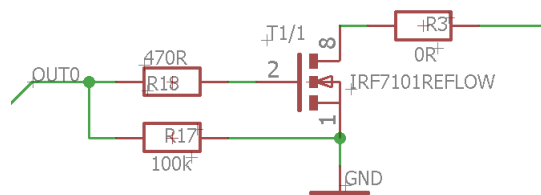
Digitální vstupy jsou realizovány dvěma způsoby. První z nich tzv. pomalé digitální jsou navrženy jako galvanicky oddělené pomocí optočlenů. Schéma viz obrázek 2.2a. Vzhledem k tomu, že v době návrhu nebylo zcela jasné konkrétní využití daného vstupu, byl tudíž na vstup připojen také přepínací jumper. Díky tomuto jumperu je

možné zvolit, zdali bude vstup optočlenu reagovat na připojení 12V nebo na spojení se zemí. Do série je také připojen rezistor, jenž omezuje proud do LED diody v optočlenu na katalogovou hodnotu. Výstup optočlenu je připojen přímo na vstup mikrokontroléru a také se zde nachází Pull up rezistor, jenž zajišťuje výchozí stav výstupu. Druhý typ zapojení tzv. rychlých digitálních vstupů je realizován pomocí děliče napětí a pro ochranu vstupu mikrokontroléru proti napětovým špičkám je zde použita BAV dioda. Ochrana pomocí optočlenů zde není možná, protože optočlen není schopen přenášet rychlé změny signálu. Tyto vstupy jsou přivedeny na vývody mikrokontroléru, kde jsou vnitřně připojeny periférie čítačů. Pak je možné použít zabudované funkce pro počítání a měření šířky pulsů.

Digitální výstupy jsou realizovány pomocí tranzistoru, jenž spíná zem do výstupního konektoru schéma viz obrázek 2.2b. Rezistor R117 zajišťuje výchozí stav.



(a) Digitální vstupy

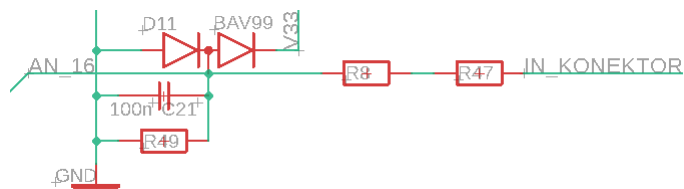


(b) Výstup

Obr. 2.2: Digitální vstupy a výstupy

## 2.6 Analogové vstupy

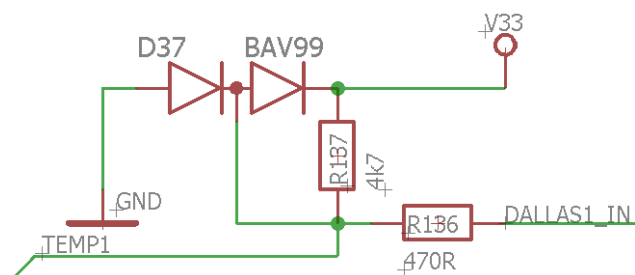
Analogové vstupy slouží například k měření napětí akumulátoru. Schéma viz Obrázek 2.3 Napětí je přivedeno na vstup děliče, který zajistí snížení napětí tak, aby bylo v rozsahu ADC převodníku tedy 0 až 3,3V. Stejně jako u digitálních vstupů je zde ochrana realizována pomocí diod.



Obr. 2.3: Analogový vstup

## 2.7 Měření teploty

Měření venkovní teploty a také teploty uvnitř vozu je zajištěno díky číslicovým teploměrům Dallas od firmy MAXIM s rozlišením 12 bitů. Teploměry jsou připojeny k mikrokontroléru pomocí sběrnice One-wire, kde každý z teploměrů je rozlišen podle unikátní adresy, která je dána od výrobce. Stejně jako u digitálních a analogových vstupů je i zde vytvořena ochrana vstupu mikrokontroléru proti napětovým špičkám. Schéma viz Obr.2.4(a). Teploměr dle datasheetu od výrobce je schopen měřit teploty v rozsahu od  $-55^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$ . Přičemž výrobce udává přesnost  $0,5^{\circ}\text{C}$  na měřeném rozsahu od  $-10^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ . Teploměr je možné zakoupit ve dvou variantách. První varianta je holá součástka s typem pouzdra TO-92, která na první pohled vypadá jako tranzistor. Tato varianta ovšem není příliš vhodná na venkovní použití. Druhá varianta je zakrytována nerezovým válečkem a tím je vhodná na venkovní použití, protože má vodotěsnou ochranu. První varianta je použita pro měření vnitřní teploty a druhá pro měření venkovní teploty.



(a) Obvod pro připojení teploměru



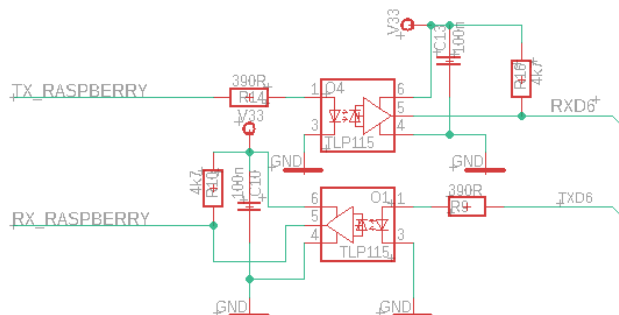
(b) teploměr DALLAS [11]

Obr. 2.4: Měření teploty

## 2.8 Komunikace s externí jednotkou

Pro komunikaci s externí Informační jednotkou je použito rozhraní UART. Komunikační linka s externí řídicí jednotkou je přivedena na vstup optočlenu TLP115 jehož

výstup připojen na vstup mikrokontroléru. Optočlen zde zajišťuje ochranu proti rušení. Schéma viz Obrázek. 2.5. Přenosová rychlost 115200/8N1 (8 datových bitů, paritní bit žádný, stop bit 1).



Obr. 2.5: Připojení externí jednotky

## 2.9 Průmyslová sběrnice RS-485

RS-485 je průmyslová sériová sběrnice. Tato sběrnice může být realizována, buď pomocí dvou vodičů nebo čtyřech vodičů. V případě čtyřech vodičů, zapojení poskytuje plně duplexní přenos bez nutnosti řízení směru přenosu. V našem případě je používáno zapojení se dvěma vodiči, kde může přenos probíhat jedním nebo druhým směrem a tento směr je nutné řídit. Připojení ke sběrnici je realizováno pomocí integrovaného obvodu ADM2687E, který převádí signály ze sběrnice a posílá je do mikrokontroléru. Na vstupu obvodu jsou realizovány přepětové ochrany, dva rezistory pro upnutí sběrnice a také se zde nachází zakončovací odpor, který je možné připojit přes jumper. Schéma viz příloha.

## 2.10 I2C

I2C je synchronní sériová sběrnice, která je realizována pomocí dvou vodičů SDA a SCL. Po vodiči SCL se posílá hodinový signál a pro vodiči SDA data. Nevýhoda této sběrnice je že oproti RS-485 není odolná proti rušení. Na této sběrnici je vždy v době vysílání jen jeden Master a další zařízení, která jsou připojená musejí být v režimu Slave.[13] Každé zařízení má svoji adresu, aby jej bylo možné rozpoznat. Klidový stav této sběrnice je, když oba signály jsou v režimu High. To je zajištěno dvěma Pull-up rezistory. Tyto rezistory jsou i v našem zapojení, kde je navíc realizována ochrana proti napětovým špičkám, jenž mohou na sběrnici proniknout. Schéma viz příloha

## 2.11 CAN

CAN je sériová datová sběrnice vyvinutá firmou Bosch. Sběrnice sama o sobě je symetrický nebo asymetrický dvou vodičový obvod, který může být odstíněný či neodstíněný. Elektrické parametry fyzického přenosu jsou specifikované normou ISO 11898.[12] CAN je realizován pomocí integrovaného obvodu ADM3054. Stejně jako u RS-485 je zde realizována přepětová ochrana. Schéma viz příloha.

## 2.12 Vlastní návrh desky plošného spoje

DPS(Deska plošného spoje) je navržena v programu Eagle, který je od společnosti Autodesk. DPS je tvořena dvěma vrstvami. Šířka cesty pro signálové spoje je 10 mil. U napájecího zdroje, kde je připojeno 12V, je šířka cest 56mil kvůli průchodu větších proudů. Spoje jsou realizovány v rastru 10mil. Pro průchody je použit otvor 0.5mm.

Na vrchní červené vrstvě se nachází většina součástek včetně mikrokontroléru a všech konektorů. Při návrhu bylo cílem, aby se konektory pro vstupy nacházely na jedné straně desky a konektory pro výstupy na protější straně z důvodu snadnějšího rozlišení. Každý ze tří typů vstupů, které byli popsány v bodech 2.5 a 2.6 jsou vyvedeny do samostatného konektoru, buď canon s devíti piny nebo s patnácti piny. Konektory canon byli použity z důvodu snadné dostupnosti, nízké ceny a jejich výhodou je, že se dají dobře zajistit šroubkem proti případnému vysunutí během jízdy v automobilu. Na vrchní straně je rozlita vrstva mědi v místě zdroje, která je připojena ke GND a ve zbylé části desky je druhá vrstva, jenž je připojena ke 3,3V. Na spodní straně se nachází obvody, které zajišťují komunikace to znamená obvody pro CAN, RS-485, komunikace s externí Informační jednotkou, komunikace s GPS. Na spodní straně se také nachází rozlita souvislá vrstva mědi, která je připojena ke GND. To bylo realizováno z důvodu odolnosti vůči rušivým signálům. Kompletní DPS viz příloha.

Tab. 2.1: Parametry DPS

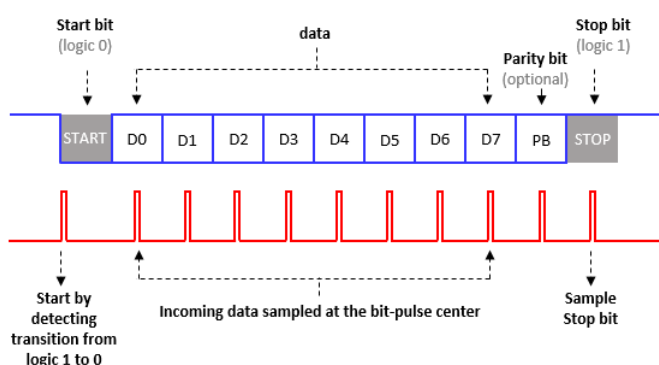
Vrstvy	2	Počet spojů	3769
Šířka signálových spojů	10 mil	Počet průchodů	226
Šířka napájecích spojů	56 mil, 40 mil	Napětí	12V, 5V, 3.3V
Vrtání průchodů	0.5mm	Počet vstupů	21
Rastr	10 mil	Počet výstupů	8
Vrstva mědi	GND, 3.3V	Rozměry desky	125 x 100 mm

## 3 Protokol a zdroj dat

Vzhledem k tomu, že došlo k návrhu protokolu pro přenos dat dříve než byl realizován hardware, bylo nutné vytvořit umělý zdroj generovaných dat. Tento zdroj je realizován pomocí počítačové formulářové aplikace napsané v programovacím jazyce C#. Na základě přesně definovaného požadavku od externí informační jednotky je prostřednictvím sériového portu, odeslána odpověď s požadovanými daty. Vzhledem k tomu, že dat je mnoho a některá je potřeba posílat v jiných časových intervalech než jiná, není možné je tudíž poslat všechny najednou. Proto byla data rozdělena do několika skupin.

### 3.1 UART

UART je asynchronní sériová sběrnice, která je použita ke komunikaci mezi informační jednotkou a tímto hardwarem pro sběr dat. Sběrnice je realizována pomocí dvou vodičů RX a TX. Pro správnou funkci musejí být všechny parametry přenosu stejné na obou stranách této sběrnice.



Obr. 3.1: UART diagram [14]

### 3.2 Protokol pro přenos dat

Komunikační protokol, který je použit pro výměnu dat mezi touto jednotkou a externí informační jednotkou, je někdy také nazýván jako binární. Binární protokol znamená, že spíše než textové řetězce, které jsou lépe čitelné pro člověka, jsou v našem případě přenášeny datové struktury. Datové struktury jsou lépe čitelné pro stroj. Při návrhu komunikačního protokolu byl kladen důraz na rychlou a efektivní výměnu dat. Protokol má odlišný formát pro požadavek a pro odpověď. Oba formáty jsou podrobněji popsány níže.

Tab. 3.1: formát protokolu pro požadavek

Délka	Číslo povelu	Parametry	CRC
-------	--------------	-----------	-----

### 3.2.1 Popis požadavku

**Délka** - délka přijaté zprávy velikost 2Byty

**Číslo povelu** - udává, která data se mají poslat, ale také lze pomocí povelu nastavit některé údaje v této jednotce 1Byte

**Parametr** - údaj pomocí něhož se předávají hodnoty v případě, že potřeba něco nastavit v této jednotce, v případě že se jedná pouze o žádost o data je tento parametr vynechán

**CRC** - Kontrolní součet všech přijatých dat. Výpočet je prováděn sečtením všech bytů. Na straně C# aplikace dojde k porovnání přijatého a vypočítaného CRC a v případě bezchybného přenosu jsou hodnoty stejné. Velikost proměnné kontrolního součtu je 4Byty

Tab. 3.2: formát protokolu pro odpověď

Délka	ACK/NACK	Číslo chyby	Data	CRC
-------	----------	-------------	------	-----

### 3.2.2 Popis odpovědi

**Délka** - délka odeslaných dat, počítá se včetně ACK a čísla chyby, velikost 2Byty

**ACK/NACK** - pokud bude přijat povel správně v odpovědi pošle se hodnota ACK

**Číslo chyby** - vrací číslo chyby, ke které došlo během přenosu nebo v zadání dotazu např. chyba CRC nebo neznámý dotaz. Pokud k žádné chybě nedošlo bude tento byte nulový 1Byte

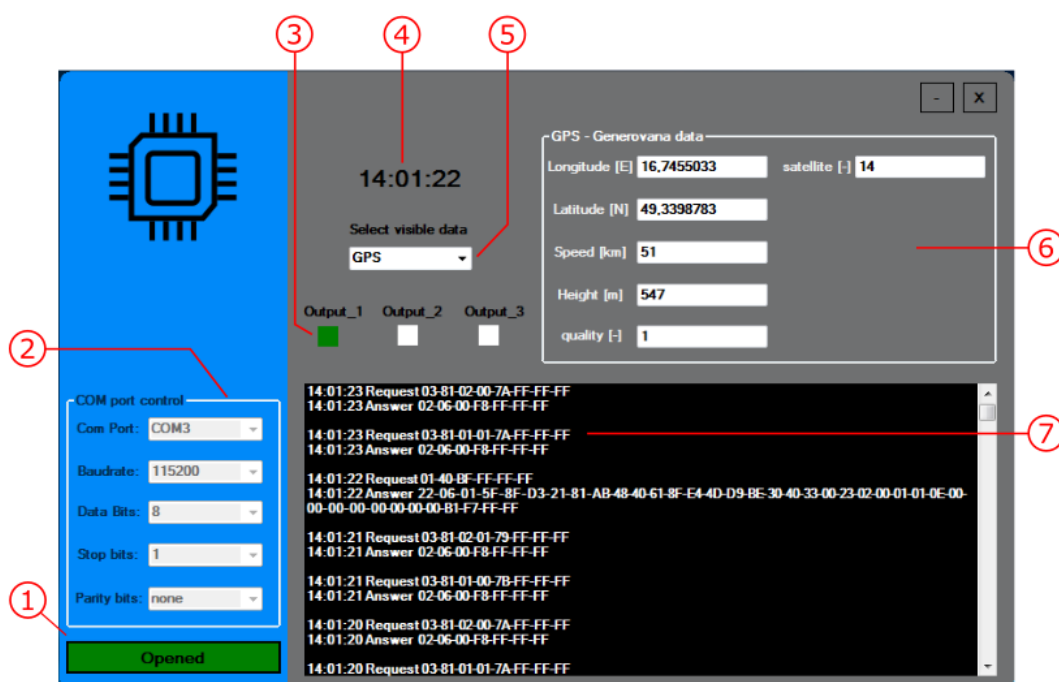
**Data** - dodatečné informace, které jsou využívány u konkrétních operací, například k nastavení případných parametrů. V případě, kdy se jedná pouze o žádost nikoli o nastavení, je tento parametr vynechán, jeho velikost závisí na konkrétním povelu

**CRC** - Kontrolní součet všech přijatých dat. Výpočet je prováděn sečtením všech bytů. Na cílové straně dojde k porovnání přijatého a vypočítaného CRC a v případě bezchybného přenosu jsou hodnoty stejné. Velikost proměnné kontrolního součtu je 4Byty



### 3.3 Umělý zdroj dat

Aplikace je spojena s externí informační jednotkou pomocí USB převodníku. Architektura vytvořené aplikace je navržena následujícím způsobem. Uživatel po spuštění aplikace nastaví parametry sériového přenosu, zvolí com port a následně provede otevření sériového kanálu. Od této chvíle je program připraven pro komunikaci s externí zobrazovací jednotkou. Většina dat v aplikaci je generována, jako náhodná čísla, která se generují každé 3 sekundy. V programu je dále pevně vloženo 20 reálných souřadnic, které jsou uloženy v poli a každé tři sekundy dochází k postupnému vyčtení hodnoty z pole. Jakmile se dojde na konec vyčítání, začíná znovu od začátku. Aplikace je napsána v prostředí Visual studio od společnosti Microsoft. Architektura aplikace je rozvržena do tří částí neboli tříd. Hlavní třída obstarává samotné zobrazení formulářové aplikace. Druhá třída, která se nazývá SerialLine, rozšiřuje vlastnosti původní systémové třídy SerialPort o kontrolu příjmu a je zde implementován vytvořený protokol. Třetí třída Servis obstarává generování dat a je volána v případě bezchybného příjmu, aby naplnila buffer daty, která mají být odeslána nazpět externí řídicí jednotce.



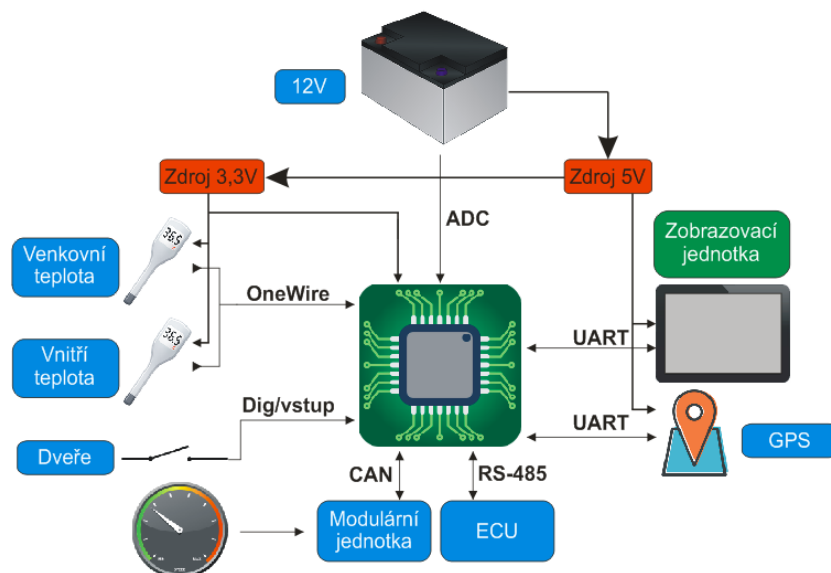
Obr. 3.2: C# aplikace jako umělý zdroj dat

### **3.3.1 Popis aplikace**

- 1** - Tlačítko pro otevření nebo zavření komunikačního portu
- 2** - Volby pro nastavení parametrů sériového portu lze měnit pokud je port zavřený
- 3** - Ikona reprezentuje výstup, který je v případě potřeby možný spínat ze zobrazovací jednotky. Pokud je ikona bílá, výstup je vypnutý, pokud je zelená, je zapnutý
- 4** - Reálný systémový čas i tento čas je v případě potřeby možné poslat na informační jednotce na vyžádání
- 5** - Rozbalovací seznam ve kterém je možné nastavit, která data mají být viditelná uživateli
- 6** - Aktuální náhodně generovaná data reprezentující GPS informace
- 7** - V okně je možné vidět probíhající nebo minulé záznamy o aktivitách.

## 4 Realizace hardwaru

### 4.1 Blokové schéma



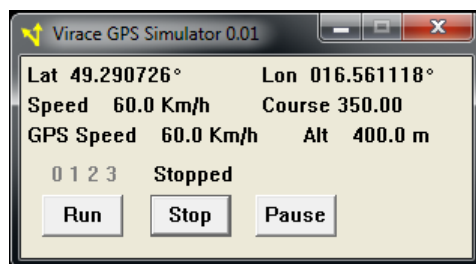
Obr. 4.1: Blokové schéma znázorňující připojení komponent k jednotce

### 4.2 GPS

Pro testovací účely rozhraní je použita počítačová aplikace Virace[15], kterou je možné vidět na obrázku 4.2. Virace simuluje modul GPS, který během testování na pracovišti není vhodný, kvůli špatnému signálu uvnitř budovy. Aplikace je spojena s jednotkou pomocí rozhraní RS232. Virace posílá data ve stejném formátu jako GPS, takže je možné zároveň otestovat funkčnost dodané knihovny pro zpracování přijatých dat.

### 4.3 Měření napětí autobaterie

Pro měření napětí autobaterie se využívá integrovaného AD převodníku. K tomuto AD převodníku je připojen odporový dělič. Protože referenční napětí AD převodníku je 3,3V, bylo nutné měřené vstupní napětí z autobaterie snížit odporovým děličem tak, aby jeho maximální měřená hodnota nepřesáhla danou referenci 3,3V. Při volbě rezistorů v děliči je uvažována maximální hodnota napětí 20 V. Potřebný dělicí poměr je tedy 1:6. Při realizaci děliče, byla také brána v úvahu skutečnost,



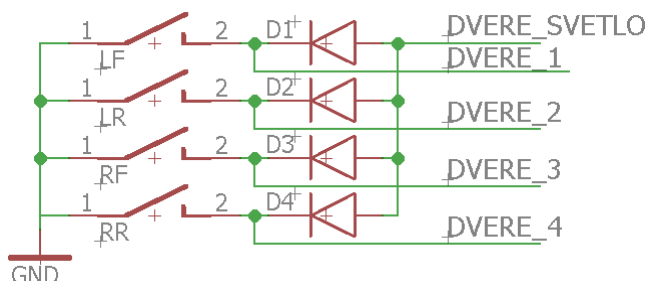
Obr. 4.2: Virace aplikace pro testování GPS knihovny a rozhraní

že požadovaná hodnota odporu nemusí být k dispozici, proto byl při návrhu plošného spoje kladen důraz, aby tuto hodnotu odporu bylo možné vytvořit kombinací dvou rezistorů. Hodnota rezistoru  $R_3$  je zvolena na 4,7 k. Následným dosazením do vztahu 4.1 pro výpočet odporového děliče získáme výslednou hodnotu druhého rezistoru 26,6 k. Tato hodnota odporu je pak následně složena ze dvou sériově řazených rezistorů 22k a 4,7k.

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} [U] \quad (4.1)$$

## 4.4 Detekce otevřených/zavřených dveří

Pro detekci stavu otevřených/zavřených dveří jsou využity digitální vstupy. Dále jsou využity koncové snímače, které již byly umístěny ve dveřích vozidla. Tyto koncové snímače jsou propojeny z již zmíněnými digitálními vstupy. Každý z koncových snímačů přivádí na digitální vstup záporný zemní pól napětí a aby se zajistilo rozsvícení diody uvnitř optočlenu, je nutné přivedení 12V na druhý kontakt diody a k tomu je využit přepínací Jumper. Na obrázku 4.3 je možné vidět původní schéma zapojení dveřních kontaktů, které je rozšířeno o diody a vyvedený výstup na každém kontaktu. Tento výstup je připojen k již zmíněnému optočlenu.



Obr. 4.3: Schéma dveřních kontaktů

## 4.5 Měření doby vstřiku paliva

Počáteční představa měření doby vstřiku spočívala v zásahu do elektroinstalace vozidla konkrétně do kabelů vedoucích ke vstřikovacímu ventilu a následné analýze signálu a provedení výpočtu spotřeby z doby vstřiku změřeného přímo na přívodním kabelu. Tento způsob se po provedeném experimentálním měření a nedostatečném počtu materiálů dokumentujících zadaný problém stal velice problematickým, proto se toto řešení ukázalo jako nevhodné. Z těchto důvodů bylo přikročeno k alternativní možnosti, kterou je vyčtení požadovaných hodnot z řídicí jednotky automobilu přes komunikačního rozhraní. Řídicí jednotka automobilu (ECU) disponuje komunikační linkou označenou jako K-LINE. Skrze toto rozhraní lze z jednotky vyčítat jednotlivé parametry, jako je například výrobní označení jednotky, typ nebo data kterými jsou například právě doba vstřiku paliva. Jednotlivé parametry jsou posílány prostřednictvím protokolu KW-1281, jenž je definován normou ISO 9141. Propojení jednotky pro sběr dat a ECU je realizováno pomocí sběrnice RS-485. Problémem ovšem je, že již zmíněné rozhraní K-LINE není kompatibilní z pohledu napěťových úrovní. Dále pak i z pohledu typu rozhraní, kde komunikace probíhá prostřednictvím jednoho vodiče namísto standardních dvou, jak tomu je u běžné sériové komunikace. Z těchto důvodů byl mezi ECU a sběrnici RS-485, prostřednictvím které probíhá komunikace, připojen převodník, jenž převádí signály z K-LINE na běžné signály sériové komunikace a také na správné TTL úrovně. Tyto signály jsou poté přivedeny na integrovaný obvod MAX485, který zajišťuje vysílání a příjem dat na sběrnici. Zároveň z důvodu nutnosti přepínání směru na vysílání a příjem je řídicí pin na MAX485 propojen s TXD pomocí invertoru. Tento způsob zapojení je zvolen kvůli snazší realizaci, tudíž není potřeba další mikrokontrolér pro řízení směru.

### 4.5.1 Nastavení spojení

Prvním problémem při implementaci bylo, že jednotka vyžaduje pro počáteční inicializaci velice pomalou komunikační rychlost od jednotky nebo programu, který s ním bude komunikovat. Komunikace musí probíhat na komunikační rychlosti 5 Baud čehož UART na mikrokontroléru není schopen. Tento problém byl odstraněn generováním potřebných sekvencí logických úrovní signálu na výstup vysílacího pinu bez využití UART periférie. Tato sekvence je v hlavním programu řešena pomocí pevně stanovených časových intervalů jedniček a nul odpovídajících komunikační rychlosti 5 Baudů jdoucích za sebou. Vzhledem k tomu, že tuto operaci nelze realizovat v součinnosti s UART periférií, je tento proces proveden při startu hlavního programu. Po ukončení inicializační sekvence je provedena korektní inicializace UART periférie na komunikační rychlost 9600 Baudů nebo 4800 Baudů podle typu ECU. Na této

komunikační rychlosti řídicí jednotka v běžném režimu předává data. Komunikace probíhá na stejných pinech, jako tomu bylo u inicializace.

Tab. 4.1: Nastavení spojení s řídicí jednotkou[16]

HWU	ECU	Popis
0x01	-	
-	0x55	Synchronizační bajt
-	0x01	Dolní významný byte LSB
-	0x8A	Horní významný byte HSB
0x75	-	Doplňěk k 0x8A

## 4.5.2 Popis protokolu KW-1281

Komunikace probíhá dvěma způsoby. Po inicializaci ECU je jednotka pro sběr dat v režimu slave a ECU v režimu master. Master vyšle znak v hexadecimálním tvaru a slave jako potvrzení vyšle doplněk tohoto čísla. To znamená např. Master odešle hodnotu 0x33 a Slave mu na tento znak vrátí odezvu na základě tohoto výpočtu  $0xFF - 0x33 = 0xCC$ . Toto probíhá tak dlouho dokud není přijat znak konce bloku od Master jednotky 0x03. Na tento poslední znak již Slave jednotka doplněkem neodpovídá. Následně se prohodí role a tato jednotka je v režimu Master a ECU v režimu slave. Master v tomto případě odesílá potvrzovací blok, který značí bezchybný příjem dat. Popis první identifikační zprávy od řídicí jednotky automobilu je v tabulce 4.5.2.

Každá zpráva obsahuje na začátku délku přijatých dat. Také obsahuje počítadlo bloků, což je bytová proměnná, jenž je inkrementována při každé výměně dat. Jak při dotazu tak při potvrzovacím paketu ACK, který bude popsán níže. Pokud proměnná dosáhne hodnoty 255 začne se počítat znovu od nuly. Typ bloku dává informace o datech jenž se nachází v přijaté zprávě (bloku). Zde jsou vypsány pouze hodnoty které se aktuálně využívají.

0x09 - příkaz ACK

0x29 - skupinové čtení dat jenž jsou žádány od ECU, jednotka umožňuje čtení pouze jedné konkrétní hodnoty. Tato možnost není využita.

0xF6 - označuje odpověď v ASCII datech, jenž obsahují například typ jednotky kterým může být SIMOS\_2P

0xF4 - označuje odpověď na žádost skupinových číselných dat

Tab. 4.2: Příklad přijatých dat od řídicí jednotky

HWU	ECU	Popis
-	0x0F	délka zprávy
0xF0	-	doplněk čísla 0x0F
-	0x03	počítadlo bloku
0xFE	-	doplněk čísla 0x03
-	0xF6	typ bloku (ASCII hodnoty)
0x09	-	doplněk čísla 0xF6
-	0x53	'S'
0xAC	-	doplněk čísla 0x53
-	0x49	'I'
0xB6	-	doplněk čísla 0x49
-	0x4D	'M'
0xB2	-	doplněk čísla 0xB2
-	0x4F	'O'
0xB0	-	doplněk čísla 0x4F
-	0x53	'S'
0xAC	-	doplněk čísla 0x53
-	0x20	'space'
0xDF	-	doplněk čísla 0x20
-	0x32	'2'
0xCD	-	doplněk čísla 0x32
-	0x50	'P'
0xAF	-	doplněk čísla 0x50
-	0x03	znak konce bloku je použit na konci každé zprávy

ACK zprávu posílá tato jednotka jako potvrzení, že přijala odeslaný blok dat. Přičemž ACK zprávy se odesílají pouze v první fázi komunikace, kdy ECU posílá po synchronizaci parametry jednotky bez vyžádání. Jakmile jsou všechny bloky zpráv s identifikací jednotky přijaty, již se ACK zprávy neodesílají.

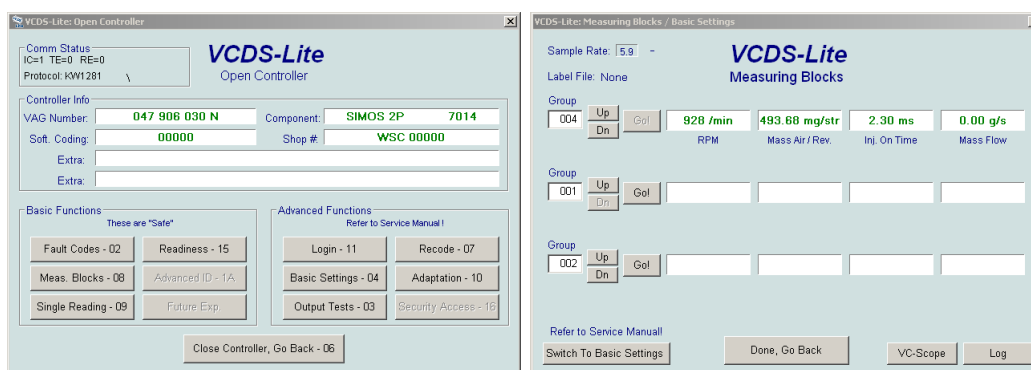
Po identifikaci jednotky jsou data z ECU posílány pouze na vyžádání. Pro snadnější vyčítání dat, jsou data rozdělena do skupin, které jsou dány výrobcem. Pro žádost na tuto skupinu je třeba odeslat dotaz do ECU s těmito parametry: délka, počítadlo bloků, příkaz pro skupinové čtení 0x29, číslo skupiny - číslováno od jedné až po x na základě typu jednotky a nakonec znak konce bloku. Odpověď na tuto žádost závisí na typu jednotky. Byli testovány dvě jednotky, jedna od firmy SIEMENS a druhá od firmy Bosch. Každá posílala odlišnou strukturu odpovědi. Jednotky totiž posílají

Tab. 4.3: Příklad ACK zprávy odeslané do ECU [16]

HWU	ECU	Popis
0x03	-	délka zprávy
-	0xFC	doplněk čísla 0x03
0x02	-	počítadlo bloku
-	0xFD	doplněk čísla 0x02
0x09	-	typ bloku
-	0xF6	doplněk čísla 0x09
0x03	-	znak konce bloku je použit na konci každé zprávy

nazpět hodnoty, které naměřili, nikoliv výslednou hodnotu, proto jsou z těmito hodnotami posílány určité parametry pro následný přepočet na požadovanou veličinu. V případě jednotky Bosch jsou tyto hodnoty posílány ve skupině spolu s daty. V případě SIEMENS jednotky jsou na dotaz nejprve odesílány parametry pro výpočet a po odeslání stejného dotazu znovu, jsou následně přijata data z konkrétními hodnotami. Proto v případě jednotky Bosch je zpráva bloku dat přibližně 2 delší než u jednotky SIEMENS.

Při analýze obsahu jednotlivých zpráv byl kromě dokumentu od výrobce, použit také diagnostický software VCDS ze stránek vag-com.cz, který je vidět na obrázku 4.4. Pomocí tohoto softwaru proběhlo první připojení k jednotce vozu a byli vyčteny první údaje. Zároveň byla během tohoto spojení, naskenována komunikace pomocí logického analyzátoru. Naskenovaná komunikace byla poté podrobena analýze, kde byl následně postupem času zjištěn význam jednotlivých hodnot, které jednotka na žádost o data posílá nazpět. Příklad naskenované a popsané komunikace jednotky Bosch viz příloha



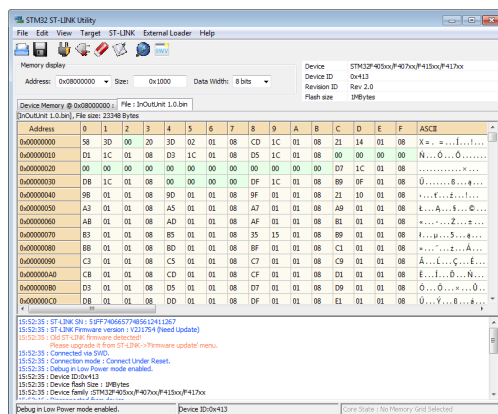
Obr. 4.4: Diagnostický software VCDS



## 5 Software

### 5.1 STM32 Utility

Tento program slouží například pro nahrávání bin souborů do mikrokontroléru, ale je také výborným prostředkem při zkoumání jestli je mikrokontrolér funkční při prvním spuštění. Po připojení ST-LINK V2 programátoru k mikrokontroléru je možné nahrávat software, editovat vnitřní paměť apod..



Obr. 5.1: ST-LINK UTILITY

### 5.2 Inicializace zařízení

Při každém spuštění jednotky dojde k inicializaci všech aktuálně používaných mikroprocesorových vstupů/výstupů, periférií a sběrnic. Dále k nastavení konkrétních rychlostí pro použité UART periferie. V případě CAN sběrnice k nastavení ID jednotlivým zprávám a u One Wire sběrnice dojde k vyhledání dostupných teploměrů na sběrnici. Dále pak dojde k odvysílání počáteční synchronizační sekvence do ECU a po té k samotné inicializaci konkrétního UARTU, použitého pro tuto komunikaci a k nastavení směru komunikace na příjem od ECU. V poslední řadě se provádí vyčtení potřebných systémových hodnot ze zálohované paměti.

### 5.3 Struktura hlavního programu

Hlavní program je realizován dvěma částmi, kde první část obstarává inicializace jednotlivých periférií a sběrnic. Tato část je popsána v bodě 5.2. Druhá část je realizována pomocí nekonečné smyčky. V této smyčce jsou cyklicky kontrolovány indikační proměnné tzv.vlajky, které indikují, jestli byla přijata nějaká data od externí

informační jednotky, modulární jednotky, ECU nebo GPS. Tyto vlajky jsou nastavovány v rámci přerušení během příjmu od konkrétní periférie nebo sběrnice, která jej vyvolala. Dále pak je v hlavním programu volána funkce, která zajišťuje uložení hodnot z AD převodníku, teploměrů, otevřených a zavřených dveří a v poslední řadě volání funkcí pro výpočet spotřeby. V programu bylo také potřeba realizovat určité časové intervaly, které zajistí cyklický sběr dat nebo pravidelné dotazování ECU na data. Tyto časové intervaly bylo potřeba realizovat tak, aby co nejméně zpomalovaly běh programu. To je zajištěno pomocí časové základny 1 milisekunda, která po uplynutí této nastavené doby vyvolá přerušení, kde jsou následně inkrementovány/-dekrementovány konkrétní časové proměnné. Jakmile konkrétní proměnná dosáhne nastavené hodnoty dojde v rámci přerušení k jejímu vynulování a zároveň k nastavení konkrétní vlajky, která značí, že bylo dosaženo nastavené hodnoty. V hlavní smyčce jsou pak pravidelně volány funkce, kde jsou jednotlivé vlajky kontrolovány a jakmile je konkrétní vlajka nastavena dojde k zahájení konkrétní operace a k vynulování vlajky.

## 5.4 Měření telemetrických hodnot

Získaná data jsou rozdělena do dvou hlavních struktur. Kde jedna struktura obsahuje telemetrická data, kromě dat získaných z GPS modulu a druhá struktura obsahuje pouze GPS data. Většina získaných dat např. teploty nebo napětí nejsou celá čísla, proto by se pro jejich uložení musel použít datový typ float u přesnějších hodnot double. V mikroprocesorové technice je ale vždy práce s desetinnými čísly náročnější na výpočetní výkon a také na paměť než s čísly celými. Z tohoto důvodu jsou kvůli menší výpočetní náročnosti, zvoleny typy proměnných jako pro ukládání celých čísel. Aby se zachovali desetinné/setinné hodnoty u získaných dat, jsou všechna čísla před uložením, vynásobena hodnotou buď deseti nebo pro větší přesnost stem. Tímto způsobem zajistíme, že nedojde ke ztrátě naměřené informace a zároveň snížíme výpočetní náročnost mikroprocesoru.

## 5.5 Rychlost a ujetá vzdálenost

Pro získání dat o aktuální, průměrné rychlosti a také počtu najetých kilometrů, je použita malá externí modulární jednotka. Modulární jednotka posílá všechna naměřená data prostřednictvím CAN sběrnice. Tyto data jsou jednak uloženy do struktury a dále pak jsou použity pro výpočet okamžité a průměrné spotřeby. Aktuální rychlost je měřena pomocí Hallový sondy, která je umístěná ze zadní strany

tachometru a při jeho otáčení během jízdy generuje pulsy. Tyto pulsy jsou následně zachyceny a převedeny na rychlost, počet ujetých kilometrů a po té odeslány na sběrnici. Tento způsob přenosu dat má výhodu, že není nutné vést vodiče od Hallový sondy přes část vozu, kde se do nich mohou indukovat různé napěťové špičky. Proto je použito pro přenos informací automobilové CAN sběrnice, která je právě pro tyto účely navržena. CAN sběrnice je obecně velmi problematická na zpracování, protože data se na ní objevují bez toho, aniž by si je nějaké zařízení vyžádalo. Příjem zprávy obstarává hardware v mikroprocesoru. Hardware obstarává filtrování ID jednotlivých zpráv na sběrnici a v případě, že nějaké ID zprávy odpovídá ID uloženému v registru, zpracuje tuto zprávu. ID jednotlivých zpráv, které chceme přijímat jsou uložena v registru. Tento registr je naplněn danými ID při startu programu. Pokud dojde ke shodě provede odeslání ACK a vygeneruje přerušení ve kterém je již zajištěno pomocí příslušných funkcí uložení přijatých informací do paměti.

## 5.6 Otevřené/zavřené dveře

V hlavním programu se při každém projití nekonečné smyčky kontroluje, zda byl pin stažen k nule na některém ze vstupů, na kterých jsou připojeny dveřní snímače. V případě, že dojde k přivedení země na vstup, to znamená vstup je na log 0, je nastaven konkrétní bit v proměnné Door, který odpovídá konkrétním dveřím. Tzn. pokud jsou dveře otevřeny je bit nastaven na hodnotu 1. V případě že jsou zavřeny pak na hodnotu 0. Pro zajištění správné funkčnosti je na vstup programově připojen vnitřní zdvihací odpor na plus tedy pull up. Který zajistí vrácení vstupu do předchozího stavu po odpojení země.

## 5.7 Napětí akumulátoru a sepnutí klíčku

Pro indikaci sepnutí klíčku je použit analogový vstup, který je realizován děličem napětí. Při přivedení napětí na tento vstup jsme schopni měřit napětí na akumulátoru a také získat informaci o sepnutí klíčku. Pro měření napětí je nakonfigurován AD převodník s rozlišením 12 bit. Toto rozlišení je zcela postačující pro naši aplikaci. Po zahájení a dokončení převodu AD převodníku je výsledná hodnota dosazena do rovnice 5.1. Výsledkem je hodnota, která se nachází na vstupu děliče. Abychom mohli tuto hodnotu uložit do celočíselné proměnné bez ztráty informace, dojde k vynásobení číslem 100 před uložením výsledné hodnoty. Následně je tato hodnota použita při kontrole zapnutí či vypnutí klíčku, kde pokud tato hodnota není nenulová značí, že klíček je zapnut.

$$U = \frac{h.Un.koef}{4095} \quad (5.1)$$

U ... hodnota napětí na vstupu [V]  
h ... výsledná hodnota vrácená AD převodníkem  
Un ... napájecí napětí mikrokontroléru  
koef ... násobící koeficient  
4095 ... maximální hodnota ADC převodníku při rozlišení 12 bitů

## 5.8 Měření teplot

Pro měření teplot v interiéru a na venkovní straně vozidla, byly použity teploměry Dallas, způsob jejich připojení byl popsán v bodě 2.7. Pro získání hodnot z těchto teploměru je použita externí knihovna. Uvnitř této knihovny je pouze dodatečně nastaven konkrétní vstup jednovodičové sběrnice One wire, na kterou jsou dané teploměry připojeny. Hodnoty z obou teploměrů jsou v rámci knihovny vyčteny a vloženy do pole, jež je následně vráceno při procesu v sběru dat v hlavním programu. V případě teploty vody, ta je přímo vyčtena z ECU vozu. A poté co je přijata, je vložena do proměnné Temp\_water v hlavní struktuře.

## 5.9 Výpočet aktuální a průměrné spotřeby paliva

Výpočet aktuální spotřeby je prováděn každou sekundu. Nejprve je zjištěno spotřebované palivo podle rovnice 5.2. Pro výpočet jsou potřeba hodnoty: doba vstřiku a počet otáček za minutu. Hodnoty jsou přijaty z ECU prostřednictvím sběrnice RS-485. Hodnota otáček je před samotným dosazením do rovnice převedena na otáčky za sekundu. Při výpočtu je předpokládáno, že počet otáček klikové hřídele je během jedné sekundy konstantní, stejně jako doba vstřiku. Při každé půl otáčky klikové hřídele je do jednoho z válců vstříknuto palivo, jehož množství je dáno dobou otevření vstřikovacího ventilu a konstantou průtoku paliva. Konstanta průtoku paliva je vyčtena ze servisních parametrů od výrobce, řádově se jedná o mikrolitry za milisekundu, konkrétní hodnota závisí na typu motoru. Tabulka 5.1 ukazuje skupinu hodnot jenž jsou získány z ECU a které jsou udány výrobcem jednotky. Údaje v prvním a třetím sloupci jsou použity pro výpočet aktuální spotřeby. Jakmile je vypočítáno množství spotřebovaného paliva za 1 sekundu, je v závislosti na rychlosti, spočítána konkrétní hodnota aktuální spotřeby podle rovnic 5.3 a 5.4. Průměrná spotřeba je počítána také každou sekundu, vždy od předchozího tankování do následujícího.

$$S = \frac{Ot.To.k.2}{60} \quad (5.2)$$

S ... Aktuální spotřeba vozidla [l/s]  
 Ot ... otáčky motoru [ot/min]  
 To ... Doba otevření vstřikovacího ventilu[ms]  
 k ... konstanta průtoku paliva [ul/ms]

Tab. 5.1: Data z ECU určeno výrobcem [18]

Skupina 2 - získaných hodnot z ECU určeno výrobcem				
Hodnota	0 ..xxx/min	1,02 - 3,06 ms	0 .. xx V	-xxx,x ...xxx,x°C
Popis	Otáčky motoru	doba vstřiku	napětí baterie	teplota vzduchu

V případě že hodnota rychlosti je větší než 10km/h, hodnota aktuální spotřeby paliva je počítána podle rovnice 5.3. Pokud rychlost klesne pod 10 km/h je hodnota aktuální spotřeby počítána podle rovnice 5.4.

$$Sa = \frac{100.s}{sp} \quad (5.3)$$

Sa ... aktuální spotřeba během jízdy [l/100km]  
 s ... ujetá vzdálenost [km/s]  
 sp ... spotřebované palivo [l]

$$Sa = Sp.3600 \quad (5.4)$$

Sa ... aktuální spotřeba během stání [l/h]  
 Sp ... spotřebované palivo [l/s]

$$Sp = \frac{100.f}{s} \quad (5.5)$$

Sp ... průměrná spotřeba [l/100km]  
 s ... ujeta vzdálenost od posledního natankování [km]  
 f ... spotřebované palivo [l]

## 5.10 GPS

Pro získání dat aktuální pozice, nadmořské výšky apod.. je použit externí GPS modul, který komunikuje s touto jednotkou prostřednictvím sériové linky přes rozhraní

Tab. 5.2: Struktura získaných a uložených dat

Formát	Proměnná	Popis
uint16_t	Act_speed	Aktuální rychlost auta
uint16_t	Av_speed	Průměrná rychlost auta
uint16_t	trip	Ujetá vzdálenost k km od zapnutí klíčku
uint16_t	time_trip	Čas od zapnutí klíčku
uint32_t	total_km	Celková ujetá vzdálenost auta
uint16_t	Fuel	Množství spotřebovaného paliva
uint16_t	Av_Consumption	Průměrná spotřeba
uint16_t	Act_Consumption	Aktuální spotřeba
int16_t	Temp_In	Teplota uvnitř vozu
int16_t	Temp_Out	Venkovní teplota
int16_t	Temp_water	Teplota vody
uint16_t	Voltage	Napětí autobaterie
uint8_t	Door	Obsahuje informace otevření/zavření všech dveří, každé dveře jsou zastupovány jedním bitem
uint8_t	Info	Informace- zapnutí/vypnutí klíčku, vlajka zajišťující vypnutí jednotky

RS-232. Vždy když přijdou data je vygenerováno přerušení ve kterém dojde ke zpracování přijatých dat. Data se zpracovávají ze dvou vět a to GGA a RMC. Kde z GGA věty jsou získány informace o aktuální pozici, nadmořské výšce apod.. a z RMC věty informace o aktuální rychlosti. Obě dvě věty jsou definovány standardem NMEA 0183. O zpracování těchto vět se stará externí dodaná knihovna, která získá z těchto vět jednotlivé hodnoty. Při práci jsou využity knihovní funkce, které při zavolání vrátí a uloží informace do příslušných proměnných. Tyto proměnné se nachází ve struktuře GPS data. Obsah této struktury je možné vidět v tabulce 5.3. V případě, že se jedná o relevantní data dojde zároveň, k synchronizaci času z GPS s časem v RTC registru. Příklad GGA a RMC věty viz níže.

```
$GPGGA,180540.615,49.3600703,N,016.7292886,E,1,07,1.0,488,M,43.5,M,0.0,0000*7D
$GPRMC,153243.0,A,49.36686,N,016.8150844,E,0.00,90.00,250520,*,*27
```

Tab. 5.3: GPS struktura získaných a uložených dat

Formát	Proměnná	Popis
double	latitude	Zeměpisná šířka
double	longitude	Zeměpisná délka
uint16_t	speed	Aktuální rychlost rychlost
uint16_t	altitude	Nadmořská výška
uint8_t	hod	Aktuální hodnota hodin
uint8_t	min	Aktuální hodnota minut
uint8_t	sec	Aktuální hodnota sekund
uint8_t	month	Číslo aktuálního měsíce v roce
uint8_t	day	Číslo aktuálního dne v měsíci
uint16_t	year	Aktuální rok
uint8_t	kvalita	Platná/Neplatná pozice
uint8_t	satelity	Počet zachycených satelitů určujících pozici
uint8_t	Azimut	Aktuální azimut
uint16_t	status	Platná/Neplatná pozice

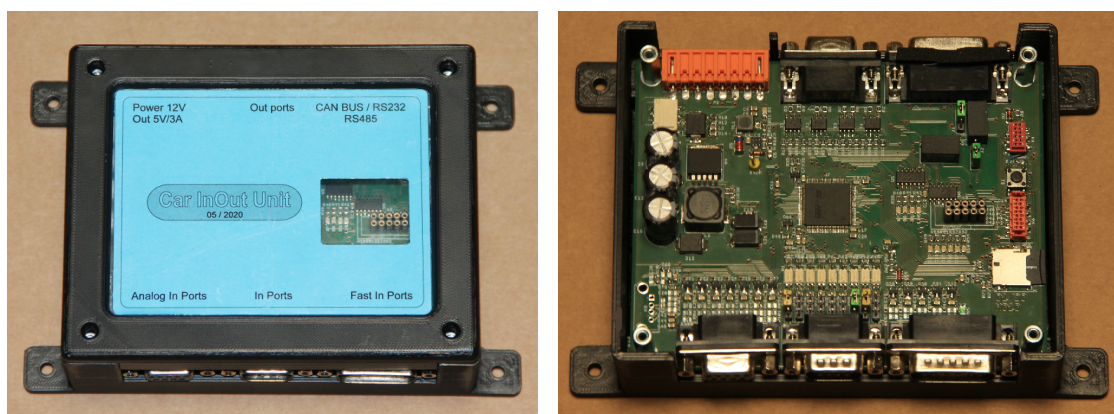
## 5.11 Komunikace s externí jednotkou

Externí informační jednotkou je vysílán dotaz na data této jednotce, při příjmu tohoto dotazu je vygenerováno přerušení, ve kterém jsou data postupně v pořadí, jak byla přijata uložena do přijímacího bufferu. Po dokončení příjmu všech dat je provedena kontrola CRC. Jestliže CRC souhlasí, nastaví se vlajka, která dává informaci, že byla přijata data a následně se ukončí obsluha přerušení. V hlavní smyčce je následně tato vlajka kontrolována a v případě, že je nastavena dojde k zavolání konkrétní funkce(povelu) na základě přijatého čísla povelu. Přehled implementovaných povelů viz tabulka 5.4 V zavolané funkci, pokud byla vyžádána data dojde k uložení dat do vysílacího bufferu. Posledním krokem je zavolání funkce, která vypočítá CRC na základě dodané délky dat a poté vloží první hodnotu, kterou je délka dat do vysílacího registru UART. V případě, že dojde k nějaké chybě nebo se jedná o neplatný povel, dojde k odeslání čísla chyby. V případě, že se jedná o povel, kde se neodesílají data, dojde k odeslání potvrzení o příjmu. Vzhledem k tomu, že tato jednotka zajišťuje zdroj energie pro externí informační jednotku je realizován postup pro vypínání celé soustavy. Tento postup je realizován takto. V hlavní struktuře jenž obsahuje data je informace o stavu klíčku. Pokud externí informační jednotka zjistí, že je klíček vypnut, vyšle této jednotce povel. Tento povel zajistí nastavení vlajky, která je kontrolována v hlavní smyčce. Pokud je tato vlajka nastavena, jednotka

provede vypnutí počítače rozepnutím relé.

Tab. 5.4: Souhrn implementovaných povelů

Povel	Parametry	Popis
P_version	bez parametru	Povel vrátí nazpět informaci o verzi firmwaru, který je nahrán v této jednotce
P_RTC_Data	bez parametru	vrátí strukturu jenž obsahuje reálné hodiny a datum
P_Car_data	bez parametru	vrátí strukturu viz tabulka 5.2 s naměřenými daty
P_GPS_data	bez parametru	vrátí strukturu viz tabulka 5.3 s naměřenými GPS daty
P_ECU_Data	bez parametru	vrátí strukturu jenž obsahuje data přijatá z ECU
P_IO	výstup a stav	povel pro sepnutí/rozepnutí daného výstupu, kde parametr 1 udává číslo výstupu, parametr 2 udává stav výstupu
P_Perm_off	bez parametru	povel nastaví vlajku která dovolí jednotce se vypnout
P_refuel	bez parametru	vynuluje spotřebované palivo



Obr. 5.2: InOut jednotka



## 5.12 Testování

V současné době bylo provedeno několik testovacích jízd, pro ověření funkčnosti implementovaných funkcionalit. Z pohledu testování jednotlivých funkcí, sběr dat z GPS modulu po celou dobu jízdy nevykazoval jakékoliv známky problému.

Dále se prokázala jako naprosto bezchybná funkce měření teplot, která po celou dobu testovacích jízd pracovala rovněž bez nejmenších potíží. Drobné chyby nastaly při komunikacích s jednotkami.

Prvotní koncept komunikace s ECU byl navržen tak, že při startu systému byla provedena inicializační sekvence a navázání spojení s ECU. Tento koncept se ukázal jako nevyhovující, po té co došlo ke krátkému přerušení jízdy a vypnutí zapalování. V tento okamžik totiž došlo k vypnutí řídicí jednotky a tím i k přerušení komunikace. Celý systém byl ovšem navržen tak, aby zůstal aktivní ještě nějakou dobu po vypnutí zapalování. V případě, že došlo k opětovnému zapnutí zapalování dříve než vypršel čas pro vypnutí hardwaru pro sběr dat, nastal problém v podobě nefunkční komunikace s jednotkou. Proto se provedla úprava stávajícího řešení, ve které se měří čas od poslední aktivity ECU. Pakliže tato doba přesáhne nastavený časový interval dvě sekundy a je zapnuté zapalování, provede se opětovná inicializace spojení s ECU. Dále bylo experimentálně zjištěno, že se komunikace nesmí přerušit na více jak 1,5 sekundy protože by došlo k výpadku aktivní komunikace s ECU a tím by vyvstala i nutnost provést opětovnou inicializaci spojení s ECU. Což je velmi časově náročné pokud chceme omezit chyby v měření spotřeby.

První návrh komunikace s informační jednotkou byl navržen tak, že dotaz od jednotky na data přicházel každé půl sekundy. Časový interval byl zvolen proto, aby data byla co možná nejpresnější. Tento návrh se ukázal také jako nevyhovující protože jednotka nestíhala přijmout všechna data a zároveň obstarávat zpracování dalších věcí jako příjem z GPS apod.. Z tohoto důvodu byla perioda dotazování od informační jednotky po dohodě zvýšena na jednu sekundu. Při tomto časovém intervalu již nevznikaly problémy se zpracováním dotazů

## 6 Závěr

V první části bylo za cíl provést rešerži dostupných jednočipových mikrokontrolérů. Při průzkumu bylo vybráno několik typů od několika výrobců a po té byl zvolen konkrétní typ, který nejlépe odpovídal požadovaným parametrům. Tímto typem je mikrokontrolér STM32F407 VGT od firmy STMicroelectronics. Následně byl s tímto mikrokontrolérem navržen a realizován požadovaný hardware pro sběr dat. Dále pak byl ve spolupráci s prací Informační jednotka pro osobní vůz zvolen binární protokol, pro výměnu dat mezi jednotkami, kvůli své úspornosti dat. Formát požadavku, který je přijímán od informační jednotky je podrobně popsán v bodě 3.2.1. Každý požadavek je potvrzován odpovědí. Formát odpovědi je popsán v bodě 3.2.2. Kvůli testování navrženého protokolu pro výměnu dat mezi jednotkami byl vytvořen umělý zdroj dat v podobě C# formulářové aplikace napsané ve Visual studiu v době nerealizovaného hardwaru. Aplikace během testování protokolu, pracovala bez nejmenších problémů. V poslední fázi bylo provedeno testování samotného hardwaru. Jednotka pro sběr dat fungovala po celou dobu z hlediska hardwaru spolehlivě. Bylo provedeno několik testovacích jízd v automobilu, aby se ověřila funkčnost i v reálných podmínkách. Pokud se objevily nějaké chyby během sběru dat nebo při jejich přenosu externí informační jednotce, jednalo se vždy o softwarovou záležitost, která byla následně vyřešena úpravou softwaru. Popis těchto chyb viz bod 5.12. Při ožívání hardwarových částí, které jsou aktuálně využívány, byli zjištěny drobné chyby, ke kterým došlo během návrhu desky. Mezi tyto chyby patří například. nepřivedená zem na programovací konektor nebo nerealizovaná ochrana proti napětovým špičkám na pinu pro měření napětí autobaterie. Co se týče programovacího konektoru, aktuálně vyřešeno drátovou propojkou přímo na desce. Ostatní zmíněné chyby byly zaznamenány a v dalších verzích hardwaru budou zapracovány do návrhu. Podněty na vylepšení do budoucích verzí jsou přivést 5V napájení do konektoru pro přívod sběrníc.

# Literatura

- [1] *NORRIS, Donald. Programming with STM32: Getting Started with the Nucleo Board and C/C++. McGraw Hill Professional, 2018.* Dostupné z URL: <https://electrovolt.ir/wp-content/uploads/2018/04/Programming-with-Stm32-Getting-Started-with-the-Nucleo.pdf>.
- [2] *Microchip Technology. SAM 4E MCUs* [online]. [cit. 13. 11. 2019]. Dostupné z URL: <https://www.microchip.com/design-centers/32-bit/sam-32-bit-mcus/sam-4e-mcus>.
- [3] *Microchip ATSAM4E-XPRO | Microchip Xplained Pro MCU Evaluation Kit* [online]. [cit. 13. 11. 2019]. Dostupné z URL: <https://mt.rsdelivers.com/product/microchip-technology/atsam4e-xpro/microchip-technology-xplained-pro-mcu-evaluation/1306161>.
- [4] *NXP Semiconductors. K20P100M100SF2 datasheet* [online]. [cit. 15. 12. 2019]. Dostupné z URL: <https://www.nxp.com/docs/en/data-sheet/K20P100M100SF2.pdf>.
- [5] *STMicroelectronics. STM32F407G-DISC1-Discovery-Kit* [online]. [cit. 7. 11. 2019]. Dostupné z URL: <https://cz.mouser.com/new/stmicroelectronics/stm-stm32f407g-disc1-kit/>.
- [6] *STMicroelectronics. STM32F407G-DISC1-Discovery-Kit datasheet* [online]. [cit. 7. 11. 2019]. Dostupné z URL: [https://www.st.com/content/ccc/resource/technical/document/user\\_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf](https://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf).
- [7] *STMicroelectronics. Nucleo 144* [online]. [cit. 7. 11. 2019]. Dostupné z URL: [https://www.st.com/content/ccc/resource/technical/document/user\\_manual/group0/26/49/90/2e/33/0d/4a/da/DM00244518/files/DM00244518.pdf/jcr:content/translations/en.DM00244518.pdf](https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/26/49/90/2e/33/0d/4a/da/DM00244518/files/DM00244518.pdf/jcr:content/translations/en.DM00244518.pdf).
- [8] *STMicroelectronics. STM32F407VGT datasheet* [online]. [cit. 20. 11. 2019]. Dostupné z URL: <https://cz.mouser.com/datasheet/2/389/stm32f405rg-956214.pdf>.

- [9] *STMicroelectronics - STM32F407VE datasheet* [online]. [cit. 20. 11. 2019]. Dostupné z URL:  
<<https://www.st.com/resource/en/datasheet/stm32f407ve.pdf>>.
- [10] *Open-source RTOS for IoT development / Mbed* [online]. [cit. 9. 11. 2019]. Dostupné z URL:  
<[https://os.mbed.com/users/hudakz/code/STM32F407VET6\\_Hello/file/1a172e2d6b86/mbed.bld/](https://os.mbed.com/users/hudakz/code/STM32F407VET6_Hello/file/1a172e2d6b86/mbed.bld/)>.
- [11] *Terra electronics - DS18B20 Waterproof Temperature Sensor Cable* [online]. [cit. 9. 11. 2019]. Dostupné z URL:  
<[https://www.terraelectronica.ru/pdf/show?pdf\\_file=%2Fz%2FDatasheet%2F1%2F1420644897.pdf](https://www.terraelectronica.ru/pdf/show?pdf_file=%2Fz%2FDatasheet%2F1%2F1420644897.pdf)>.
- [12] VAN HERREWEGE, Anthony; SINGELEEE, Dave; VERBAUWHEDE, Ingrid. *CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus. In: ECRYPT Workshop on Lightweight Cryptography. 2011.* [online]. [cit. 20. 4. 2020] Dostupné z URL:  
<<https://www.esat.kuleuven.be/cosic/publications/article-2086.pdf>>.
- [13] KIM, Yong-Jae. *I2C communication system and method enabling bi-directional communications. U.S. Patent Application No 11/028,319, 2005.* [online]. [cit. 28. 12. 2019]. Dostupné z URL:  
<<https://patents.google.com/patent/US20050165989A1/en/>>.
- [14] *Soliton Technologies - UART Protocol* [online]. [cit. 27. 10. 2019]. Dostupné z URL:  
<<https://www.solitontech.com/wp-content/uploads/2018/09/Soliton-UART-Protocol-Validation-Master-Slave-Byte-Packet-Format.png>>.
- [15] *Simulátor Virace* [online]. [cit. 20. 4. 2020]. Dostupné z URL:  
<[http://www.gpspassion.com/forumsen/topic.asp?TOPIC\\_ID=114933](http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=114933)>
- [16] Schäffer, Florian. *OBD KW 1281 Protokoll. OBD KW 1281 Protokoll.* [online]. [cit. 1. 6. 2020]. Dostupné z URL:  
<[https://www.blafusel.de/obd/obd2\\_kw1281.html](https://www.blafusel.de/obd/obd2_kw1281.html)>
- [17] Škoda.Felicia, *Vstříkovací a zapalovací zařízení Simos 2P. Servisní služby. Technické informace Vydání 10.96.* [online]. [cit. 10.04.2020]. Dostupné z URL:  
<<https://docplayer.cz/10395487-Felicia-vstoikovaci-a-zapalovaci-/>>

zaoizeni-simos-2p-servisni-sluzby-technicke-informace-500-5216-00/-15-vydani-10-96-j.html>.

- [18] Škoda. *Felicia, Dílenská příručka - Vstřikovací a zapalovací zařízení Bosch Monotronic. Technické informace Vydání 11.94.* [online]. [cit. 15.4.2020].

Dostupné z URL:

<[https://issuu.com/marty\\_fk/docs/dp\\_fel\\_motor\\_elektronika\\_mono\\_motronic](https://issuu.com/marty_fk/docs/dp_fel_motor_elektronika_mono_motronic)>.

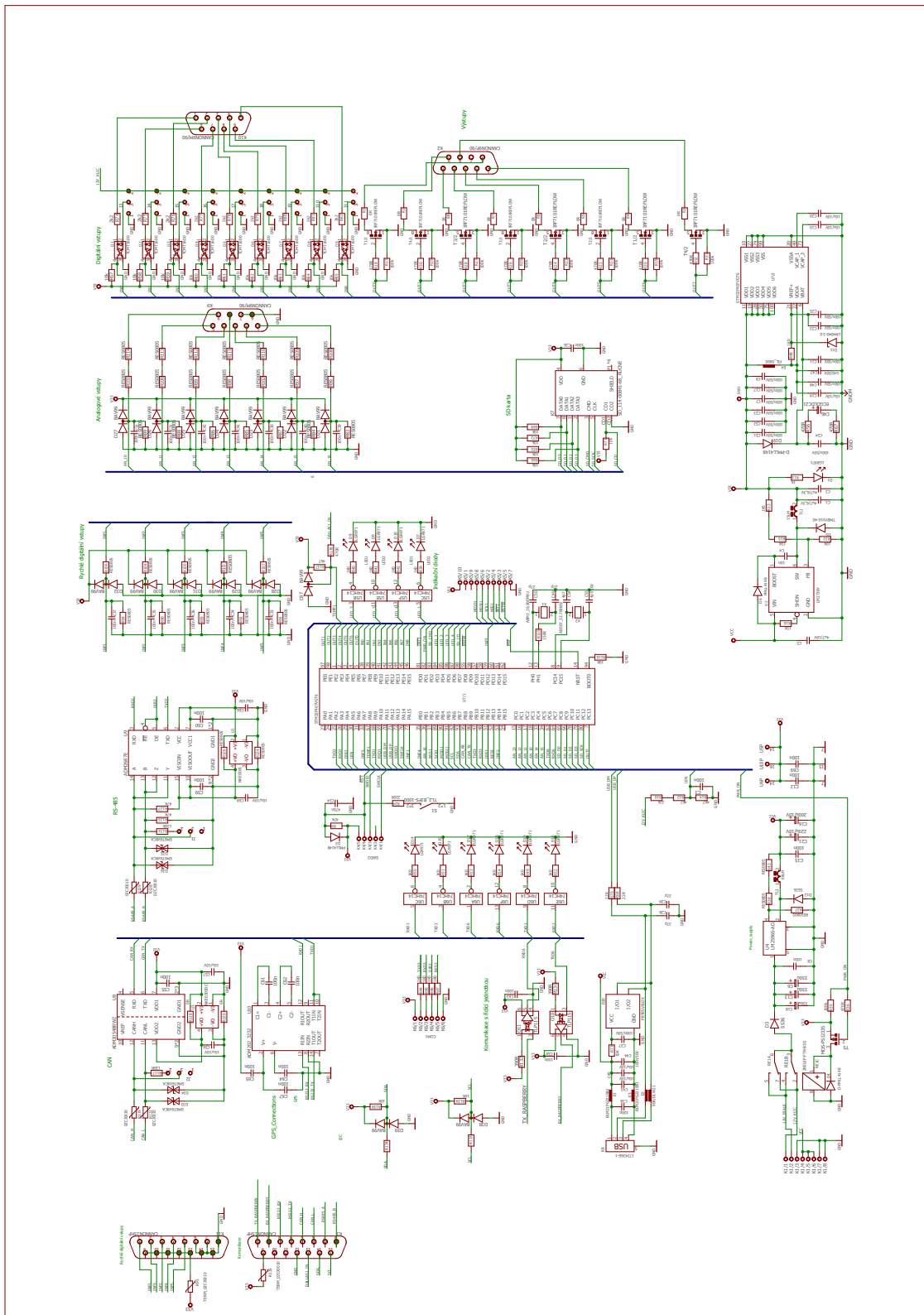
## Seznam symbolů, veličin a zkratk

<b>GPS</b>	Global Positioning System - Globální Polohovací Systém
<b>UART</b>	Universal asynchronous receiver and transmitter- univerzální asynchronní přijímač a vysílač
<b>CAN</b>	Controller Area Network
<b>RTC</b>	Real time clock - hodiny reálného času
<b>DPS</b>	Deska plošného spoje
<b>ACK</b>	acknowledgement code - potvrzovací kód
<b>NACK</b>	negative acknowledgement code - negativní potvrzovací kód
<b>CRC</b>	cyclic redundancy check
<b>USB</b>	Universal Serial Bus - univerzální sériová sběrnice
<b>ECU</b>	Electronic Control Unit - řídicí jednotka automobilů

# Seznam příloh

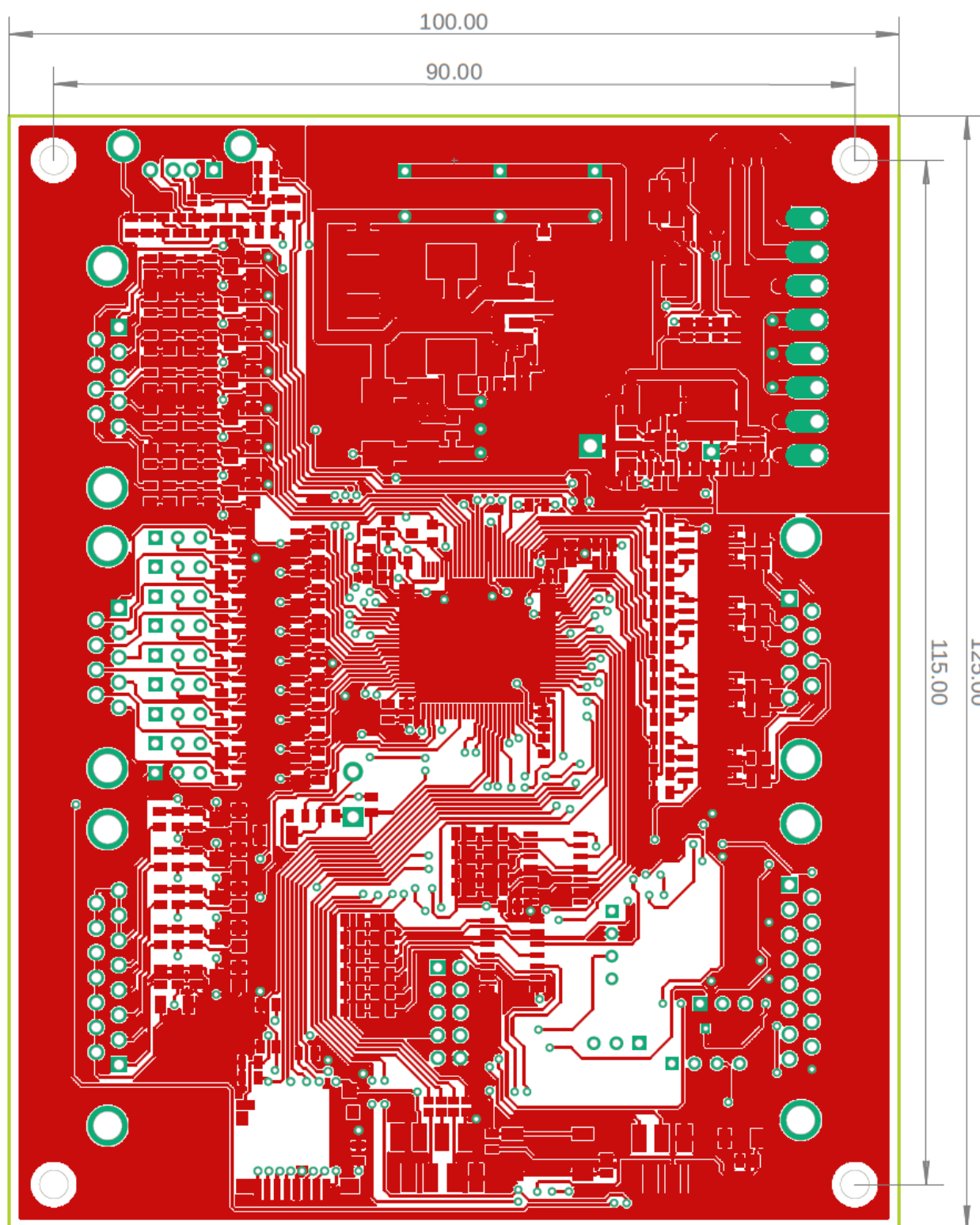
A	Schéma hardwarové desky pro sběr dat	47
B	Návrh PCB - Vrchní strana	48
C	Návrh PCB - Spodní strana	49
D	Realizovaná deska	50
E	Naskenovaný průběh odpovědi jednotka Bosch	51
F	Obsah přiloženého CD	52

# A Schéma hardwarové desky pro sběr dat

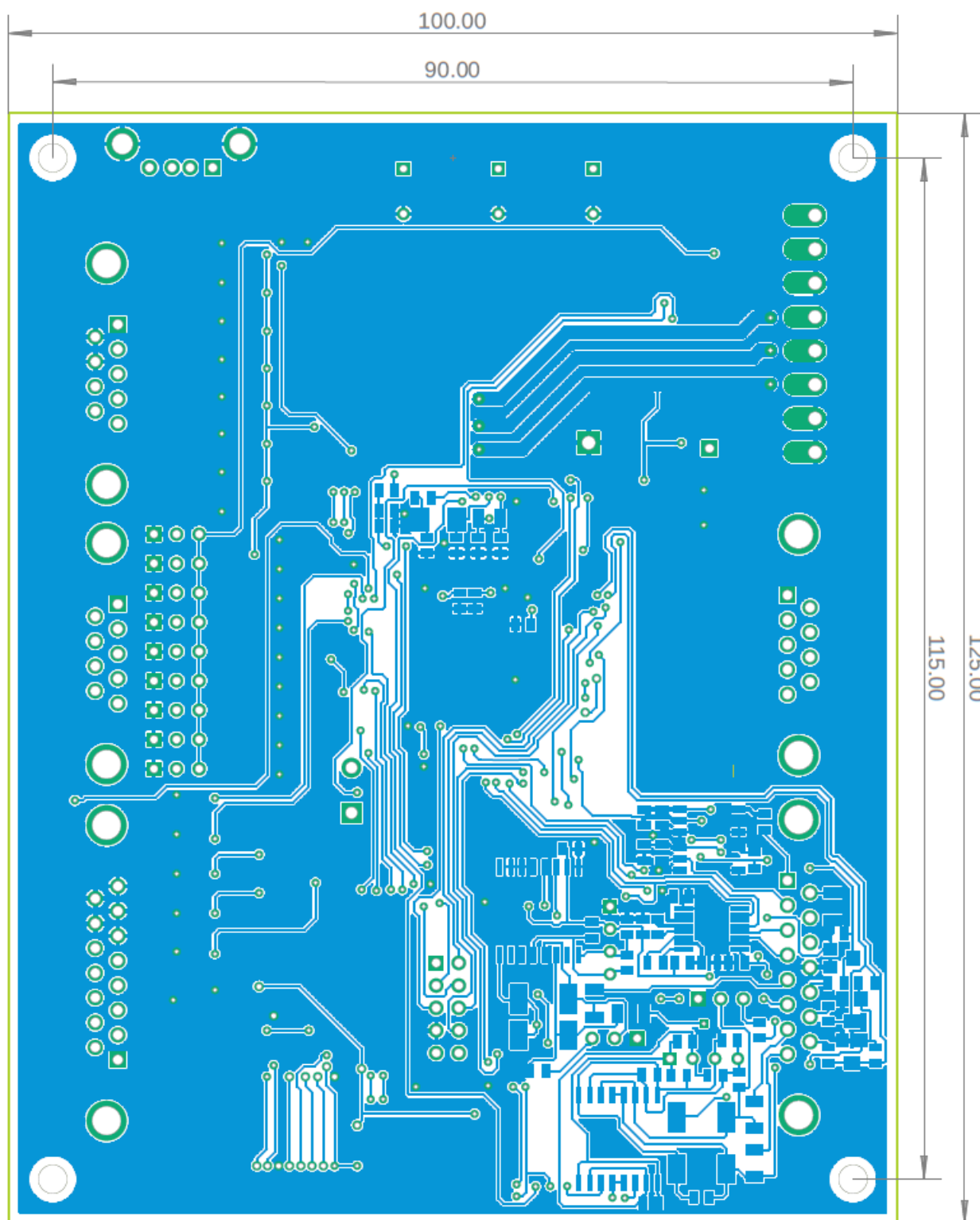




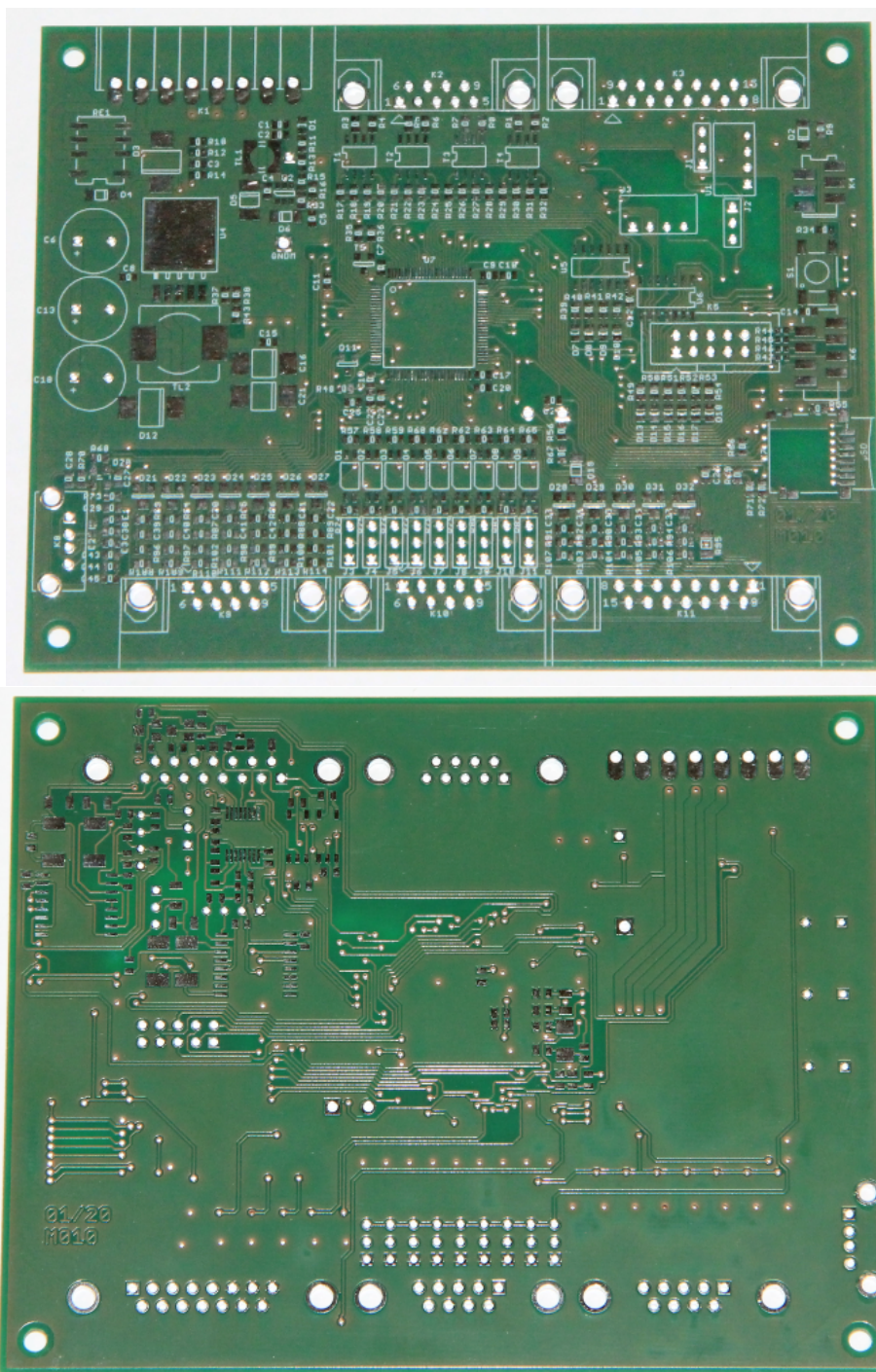
## B Návrh PCB - Vrchní strana



## C Návrh PCB - Spodní strana



## D Realizovaná deska



# E Naskenovaný průběh odpovědi jednotka Bosch



**0xF0 - délka zprávy**

0xF0 - doplněk délky

**0x07 - číslo vyměněného paketu**

0xF8 - doplněk 0x07

**0xE7 - odpověď na žádost o čtení ve skupině**

0x18 - doplněk 0xE7

**0x01 - číslo skupiny**

0xFE - doplněk 0x01

**0x7D - dekadický 125 - RPM**

0xB2 - doplněk 0x7D

**0x27 - dekadický 39- RPM**

0xD8 - doplněk 0x27

**0x05 - dekadický 5 - RPM**

0xFA - doplněk 0x05

**0x09 - dekadický 9 - Teplota vody**

Otáčky motoru(RPM) =  $0x27 * (0x7D/5)$

Teplota vody =  $(0x8C - 100) * 0x09$

Lambda napětí =  $0x01955 * 0xB0$

0xF6 - doplněk 0x09

**0x8C - dekadický 140 - Teplota voda**

0x73 - doplněk 0x8C

**0x0B - dekadický 11 - Teplota voda**

0xF4 - doplněk 0x0B

**0x27 - dekadický 39 - Lambda**

0xD8 - doplněk 0x27

**0xB0 - dekadický 176 - Lambda**

0x7F - doplněk 0xB0

**0x10 - dekadický 16 - Lambda**

0xEF - doplněk 0x10

**0xFF - dekadický 255**

0x00 - doplněk 0xFF

**0x42 - dekadický 66**

0xBD - doplněk 0x27

**0x03 - konec bloku**

## F Obsah přiloženého CD

```
/ ..... kořenový adresář přiloženého CD
├── eagle.....inout jednotka soubory
│   ├── in_out.brd
│   └── in_out.sch
├── obrazky.....ostatní obrázky
│   ├── naskenovaný průběh Bosch.png
│   ├── horní vrstva.png
│   └── spodní vrstva.png
├── moduley.....zdrojové soubory inout jednotka
│   ├── ADC.c
│   ├── ADC.h
│   ├── car.c
│   ├── car.h
│   ├── kw1281.c
│   ├── kw1281.h
│   ├── main.c
│   ├── main.h
│   ├── RPI.c
│   ├── RPI.h
│   └── SysTick.c
└── Umělý zdroj dat.....Projekt Visual studio 2017
    ├── Car_unit_simulator
```